E-LAB

Medical Diagnoses Using AI

Special thanks for our great supervisor who supported us along the way

Supervised by

Dr / Noha Hamza

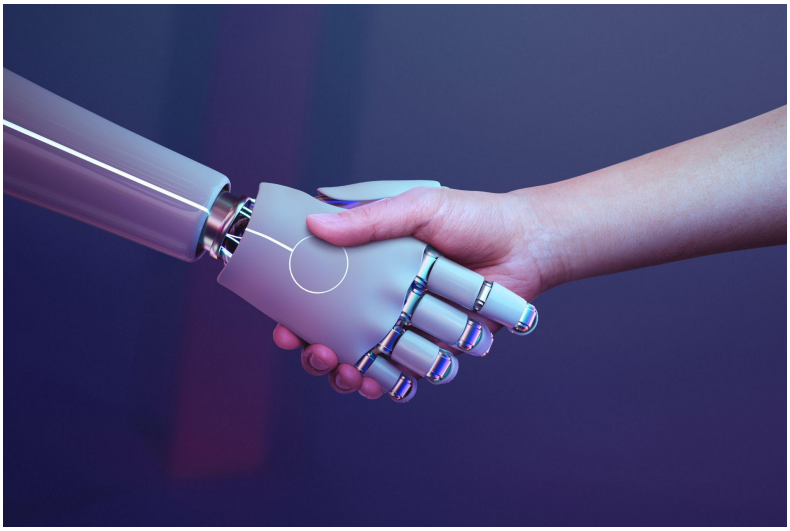
content

Chapter 1

Introduction

1.1 Introduction

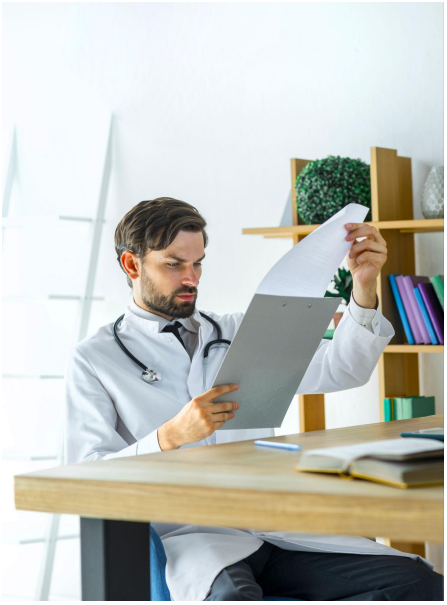
As technology continues to advance, AI started raising as a great tool to help humankind, for its great capabilities scientists thought about using it in all aspects of life.



With the capabilities of AI, scientists were able to tune it to any field they wanted, it doesn't matter how vast or hard to predict a field is, scientists always found a way to use it.

helped us in many different fields; in industry, scientific research, and even entertainment.

The field of medicine was not left behind. The use of artificial intelligence (AI) in medical diagnosis has become increasingly popular and has revolutionized the way doctors diagnose and treat patients. In this book, we will explore a software application that performs a medical diagnosis for some diseases using AI.



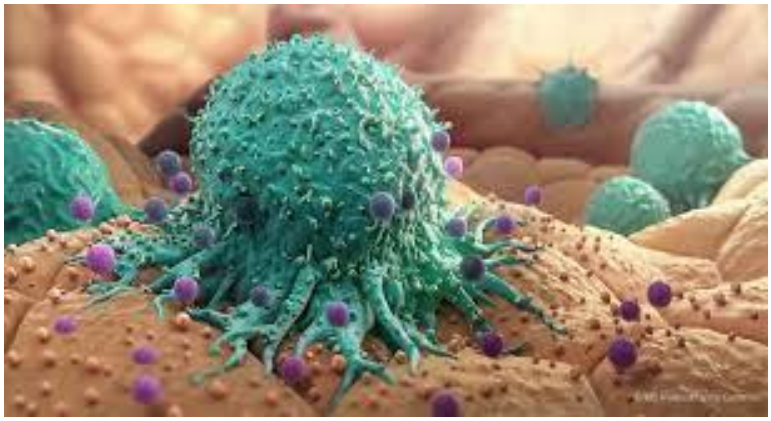
There are many diseases that can be diagnosed by using different AI models, yet in this software we will address two of the most dangerous diseases, which are cancer and diabetes.

It's a priority to do a lot of research about how to diagnose These two diseases earlier than other diseases, as it'll give the patient more time for treatment and a higher percentage of success.

1.2 Addressed Diseases

Diseases like Cancer and diabetes are two of the most prevalent and devastating health conditions affecting people worldwide. These diseases have a significant impact on the overall health and well-being of individuals, families, and communities, and can lead to severe physical, emotional, and financial consequences.

Cancer is a group of diseases characterized by the uncontrolled growth and spread of abnormal cells in the body. The disease can originate from any part of the body and can spread to other organs through a process called metastasis.



There are over 100 different types of cancer, including breast cancer, lung cancer, prostate cancer, and colon cancer, among others. According to the World Health Organization (WHO), cancer is the second leading cause of death globally, accounting for an estimated 9.6 million deaths in 2018.

Diabetes, on the other hand, is a chronic condition that occurs when the body is unable to produce or properly use insulin, a hormone that regulates blood sugar levels.

There are two main types of diabetes: type 1 diabetes, which occurs when the body's immune system attacks and destroys the insulin-producing cells in the pancreas, and type 2 diabetes, which is caused by a combination of genetic and lifestyle factors. According to the International

Diabetes Federation, diabetes affects over 463 million adults worldwide, and the number is expected to rise to 700 million by 2045.

Both cancer and diabetes are complex diseases with a multifactorial etiology, making them challenging to prevent and treat. The risk factors for cancer and diabetes include genetic predisposition, lifestyle factors such as tobacco use, unhealthy diet, lack of physical activity, and exposure to environmental toxins.

The impact of cancer and diabetes on an individual's health and well-being is profound. Cancer can cause a range of physical symptoms, including pain, fatigue, weight loss, and changes in appetite. The side effects of cancer treatments such as chemotherapy and radiation therapy can also lead to further physical complications. In addition to physical symptoms, cancer can also have a significant impact on an individual's emotional and mental health, causing anxiety, depression, and stress.

Similarly, diabetes can have a significant impact on an individual's health and well-being. Uncontrolled diabetes can lead to a range of complications, including cardiovascular disease, kidney failure, blindness, and nerve damage. These complications can significantly reduce an individual's quality of life and lead to further health complications.

In addition to the impact on individuals, cancer and diabetes also have significant social and economic consequences. The cost of cancer treatments, including surgery, chemotherapy, and radiation therapy, can be prohibitively expensive, leading to financial strain for individuals and families. Similarly, the cost of managing diabetes, including medications, monitoring equipment, and regular medical check-ups, can also be a significant financial burden.

In conclusion, cancer and diabetes are two of the most prevalent and devastating health conditions affecting people worldwide. These diseases have a significant impact on the overall health and well-being of individuals, families, and communities, and can lead to severe physical, emotional, and financial consequences.

1.3 History of using AI in the medical field

Artificial intelligence (AI) has been used in the medical field for several decades, but its use has significantly increased in recent years due to advancements in machine learning, natural language processing, and computer vision. AI has the potential to revolutionize healthcare by improving diagnosis accuracy, reducing medical errors, and providing personalized treatment recommendations. In this article, we will provide an overview of the history of using AI in the medical field.

1950s to 1970s: Early AI Applications in Medicine

The earliest use of AI in medicine can be traced back to the 1950s when researchers began using computers to simulate human thought processes. In 1956, the Dartmouth Conference marked the birth of AI as a field of study. The conference included discussions on the potential applications of AI in various fields, including medicine.

In the 1960s, researchers began developing expert systems, which were computer programs that could make decisions based on a set of rules and knowledge. One of the earliest expert systems in medicine was MYCIN, developed in the 1970s at Stanford University. MYCIN was designed to assist physicians in diagnosing bacterial infections and recommending antibiotics.

1980s to 1990s: Advancements in AI and Medical Imaging

In the 1980s, AI technology advanced rapidly, and researchers began using neural networks to improve pattern recognition and predictive modeling. Neural networks are computer systems that can learn from data and improve their accuracy over time. This technology was used in medical imaging, particularly in analyzing radiological images, such as X-rays, CT scans, and MRIs.

One of the significant developments in medical imaging was the use of computer-aided diagnosis (CAD) systems, which used AI algorithms to help radiologists detect and diagnose abnormalities in medical images.

In the 1990s,

researchers began using AI in medical decision-making and clinical decision support systems (CDSS). CDSS were designed to assist healthcare providers in making clinical decisions by providing evidence-based recommendations for diagnosis, treatment, and management of medical conditions.

2000s to Present: Personalized Medicine and Health Monitoring

In the 2000s,

AI technology continued to advance, and researchers began using machine learning algorithms to develop predictive models for personalized medicine. Personalized medicine involves tailoring medical treatment to individual patients based on their genetic makeup, lifestyle, and medical history. AI-based predictive models can be used to identify patients who are at risk for certain diseases and recommend personalized treatment options.

Another significant development in the use of AI in medicine is the use of health monitoring devices, such as wearables, to collect data on patients' health status. These devices can collect data on vital signs, physical activity, sleep, and other health-related metrics. AI algorithms can be used to analyze this data and identify patterns and trends that can be used to improve diagnosis and treatment.

AI is also being used to develop chatbots and virtual assistants in the medical field. These AI-powered systems can answer patient queries, provide health advice, and even diagnose medical conditions based on symptoms.

1.4 Applications of AI in the Medical Field

AI has several potential applications in the medical field, including:

1. Medical Imaging: AI algorithms can be used to analyze medical images, such as X-rays, CT scans, and MRIs, to identify abnormalities and assist radiologists in making accurate diagnoses.

2. Clinical Decision Support Systems: AI-powered systems can analyze patient data, such as medical history and diagnostic test results, to provide healthcare providers with evidence-based recommendations for diagnosis, treatment, and management of medical conditions.

3. Personalized Medicine: AI algorithms can be used to analyze patient data, such as genetic makeup, lifestyle, and medical history, to develop personalized treatment plans tailored to individual patients.

4. Health Monitoring: Wearables and other health monitoring devices can collect data on patients' health status, which can be analyzed using AI algorithms to identify patterns and trends and provide insights into patients' health.

5. Chatbots and Virtual Assistants: AI-powered chatbots and virtual assistants can provide patients with health advice, answer queries, and even diagnose medical conditions based on symptoms.

1.5 Challenges and Limitations of AI in Medicine

While AI has the potential to revolutionize healthcare, there are several challenges and limitations that must be addressed to ensure its safe and effective use.

1. Data Quality: AI algorithms require large amounts of high-quality data to learn and make accurate predictions. However, healthcare data is often incomplete, inconsistent, and unstructured, which can lead to inaccurate predictions and diagnoses.

2. Bias: AI algorithms can be biased if the data used to train them is biased. For example, if the data used to train an AI algorithm is biased against a particular race or gender, the algorithm may make inaccurate predictions or diagnoses for individuals belonging to that race or gender.

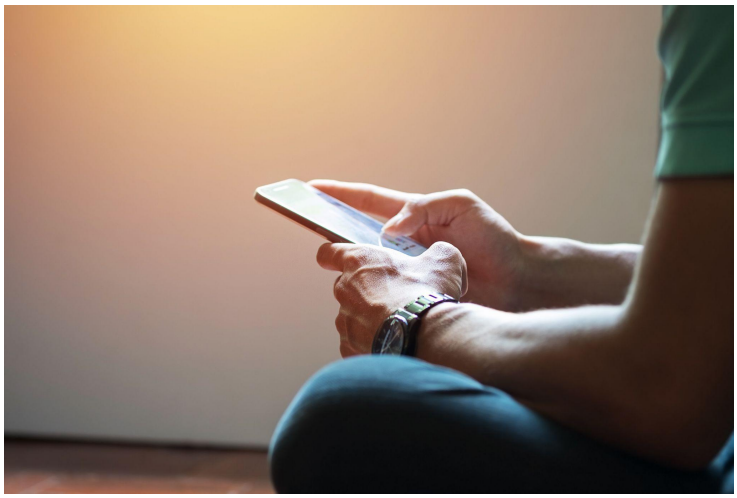
3. Regulation: The use of AI in healthcare is subject to regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). These regulations require healthcare organizations to ensure the privacy and security of patient data, which can be challenging when using AI.

4. Transparency: AI algorithms can be complex and difficult to understand, which can make it challenging to explain how they arrived at a particular diagnosis or recommendation. This lack of transparency can lead to distrust among patients and healthcare providers.

5. Ethical Concerns: The use of AI in healthcare raises ethical concerns, such as the potential for AI to replace human healthcare providers or the use of AI to make life or death decisions.

Conclusion

The use of AI in the medical field has a long history, and its potential for improving healthcare is significant. AI algorithms can be used to improve diagnosis accuracy, reduce medical errors, and provide personalized treatment recommendations. However, the use of AI in healthcare is subject to challenges and limitations, such as data quality, bias,regulation, transparency, and ethical concerns. Addressing these challenges and limitations is crucial to ensuring the safe and effective use of AI in healthcare. As AI technology continues to advance, it is likely that we will see even more significant developments in the use of AI in the medical field, potentially transforming healthcare as we know it.



The software application is designed to analyze patient data and provide as many accurate diagnoses as possible based on the symptoms and data entered. It uses machine learning algorithms to learn from vast amounts of data and improve its accuracy over time. The application can be used by both healthcare professionals and casual users to make preliminary predictions about a patient's condition.

One of the key benefits of using AI in medical diagnosis is its ability to process large amounts of data quickly and accurately. This means that doctors can also use it to get fast and tentative predictions about patient conditions, leading to better outcomes for patients. Additionally, the software application can help reduce errors caused by human error or bias.

As mentioned before, even casual users can use the application to get a simple idea about their possible condition even before visiting the physician.

The software application is easy to use and can be integrated into existing healthcare systems seamlessly. Healthcare professionals can access the application through a web-based interface or a mobile app, making it convenient for them to use on-the-go.

In this book, we will explore how the software application works and its impact on healthcare, what problems we address and how we solved them. We will also discuss some of the challenges associated with using AI in medical diagnosis and how they can be addressed.

Overall, this book provides an in-depth look at a software application that performs medical diagnosis using AI. It highlights the benefits of using AI in healthcare and explores some of the challenges associated with its implementation. Whether you are a healthcare professional or simply interested in learning more about this exciting technology, this book is sure to provide valuable insights into the future of medicine.

1.6 Problems and Solutions

The advancement that we have nowadays in technology haven't yet prevented human error, the need to rest, or the rapid need of new research, especially in the medical field.
But using such advanced technologies will reduce the chance of that error, work 24/7 without any need to rest, and will increase the speed of research. and one of these technologies was AI.

With a fairly high accuracy of results and a variety of AI types and uses, using AI is perfect nowadays to solve most of the errors caused by human hands.

But before discussing human errors and how to solve them using AI, this technology will open a big opportunity to solve a very itching problem consists of the following:

Problems

- Cancer symptoms are famous and easy to discover, but not all symptoms are easily diagnosed to be cancer, and some of the symptoms look normal and similar to other diseases.

  Same as cancer, Diabetes also might not be easily discovered as its symptoms might be considered normal activities, like (going to the bathroom multiple times a day, drinking a huge amount of water daily, changing in weight either by increasing or decreasing,...etc).

  So, It’s hard to identify both cancer and diabetes without any professional examination which takes a long time and is sometimes very expensive to some individuals, so not all people have the privilege to take the examination at any time.

- Both diseases have a very dangerous aspect about them which is that both of them might be inherited from family members, and it doesn't matter if they were close relatives like your parents or even your great grandparents.

  Which makes it more important to individuals that their family has some history with those diseases to have a rapid and periodic examination to these diseases.

- Doctors may need help to quicken the process of diagnoses, the process of identifying both cancer and diabetes takes a fairly long time and costs a lot.
  To have a tool that quickens this process and costs nothing might save a lot of patients from suffering a painful and long experience with the disease, as it’s better to discover them early to increase the percentage of treatment success.

Objective

- Providing a user-friendly UI will encourage the user to use the application more often, as it will be easy to choose the diseases, enter diagnose data, and read results.

- Providing a 24/7 available and free application to periodically check family members who experienced a history with the mentioned diseases.
  The percentage of Early Diagnosing of both cancer and diabetes will increase significantly, and more lives will be saved.

- Providing the physicians who are specified in cancer and diabetes fields a technology that can assist them in rapidly diagnosing new medical cases with a fraction of time compared to the normal process of diagnosis.
  This means more patients will be treated and more cases will be discovered early.

Chapter 2

System Analysis
&
Design

2.1 System Analysis

In today's fast-paced and ever-changing business environment, organizations rely heavily on information systems to manage their operations and make informed decisions. A well-designed information system can automate business processes, increase efficiency, and improve decision-making. However, the development of an information system is a complex process that requires careful planning, analysis, and design.

System analysis and design is a process of developing information systems that meet the needs of the users and the organization. The process involves studying the current system, identifying its weaknesses and inefficiencies, and designing a new system that addresses these issues. The system analysis and design process includes several phases, each of which is essential to the successful development of an information system.

In This book we will provide a comprehensive overview of system analysis and design in “E-LAB”. We will explain the importance of system analysis and design and the benefits that can be gained from a well-designed information system. We will also discuss the various phases of the system analysis and design process, the tools and techniques used in each phase, and the challenges and considerations that need to be taken into account during the development of an information system.

Also we will provide some diagrams related to system analysis and design that will provide an overview on how the software interacts internally (between software components and each other) and externally (between the system components and user).

2.2 Importance of System Analysis and Design

A well-designed information system can provide many benefits to an organization. It can automate business processes, reduce errors, increase efficiency, and improve decision-making. It can also provide timely and accurate information to the users, which can help them make informed decisions.

On the other hand, a poorly designed information system can result in inefficiencies, errors, and delays. It can lead to wasted time and resources, and can ultimately harm the organization's ability to compete in the marketplace.

System analysis and design is important because it ensures that the information system meets the needs of the users and the organization. It helps to identify the strengths and weaknesses of the current system and provides a roadmap for the development of a new system that addresses these issues. It also helps to ensure that the new system is designed to be scalable, flexible, and able to adapt to changing business needs.

The benefits of a well-designed information system can be significant. It can improve the accuracy and timeliness of information, reduce errors, improve efficiency, and increase productivity. It can also provide a competitive advantage by enabling the organization to respond quickly to changes in the marketplace.

2.3 Phases of System Analysis and Design

The system analysis and design process consists of several phases, each of which is essential to the successful development of an information system. The four main phases of the system analysis and design process are:

- Planning and Analysis
- Design
- Implementation

Each one of these phases is discussed in detail in the following :

2.3.1 Phase 1: Planning and Analysis

The planning and analysis phase is the first step in the system analysis and design process. This phase involves identifying the problem that needs to be solved, determining the goals and objectives of the new system, and analyzing the existing system to identify its strengths and weaknesses.

The first step in this phase is to conduct a feasibility study to determine whether the new system is technically, economically, and operationally feasible. This involves analyzing the current system, identifying the problems and inefficiencies, and determining whether a new system is necessary to solve these issues.

So starting with the feasibility study:

The study will assess the technical, economic, and operational feasibility of the software

Technical Feasibility:
The development of an AI-based cancer and diabetes diagnosis software requires a high level of technical expertise in artificial intelligence and machine learning.

Economic Feasibility:

The development of an AI-based cancer and diabetes diagnosis software will require significant financial investment. The project will require the purchase of high-performance computing hardware, software development tools, and data storage facilities. The software will also require ongoing maintenance and updates to ensure it remains accurate and up-to-date with new medical research and developments. The software's revenue model will be based on charging healthcare providers and patients for the use of the software.

Operational Feasibility:

The operational feasibility of the software will depend on the integration of the software into the current healthcare system. The software will need to be integrated with existing electronic health record systems to ensure seamless data sharing and patient care coordination. The software will also require trained medical professionals to interpret the results and provide appropriate recommendations for patient care. Therefore, the development team will need to work closely with healthcare providers to ensure the software's seamless integration into their workflow and address any potential concerns they may have.

Conclusion:

Based on the assessment of technical, economic, and operational feasibility, the development of an AI-based cancer and diabetes diagnosis software is feasible. However, the project will require a significant financial investment and a highly skilled development team to ensure the software's accuracy and compliance with medical regulations. The project's success will also depend on its seamless integration into the current healthcare system and collaboration with healthcare providers to ensure patient safety and quality of care.

Next we will be discussing the problems that this software is trying to solve:

We stated before the problems that we aim to solve by providing our software, and you can find that in the page 14

But to summarize it in few point as follows :

- Some symptoms of cancer and diabetes looks normal and not easily diagnosed to be caused by a disease
- Both diseases have a very dangerous aspect about them which is that both of them might be inherited from family members
- Doctors may need help to quicken the process of diagnoses, the process of identifying both cancer and diabetes takes a fairly long time and costs a lot.

By stating out the problems, now we will discuss the solutions that this application will serve :

- Providing a user-friendly UI will encourage the user to use the application more often, and navigate through the application, it will be easy to choose the diseases, enter diagnose data, and read results.

- Providing a 24/7 available and free application.

- The percentage of Early Diagnosing of both cancer and diabetes will increase significantly, and more lives will be saved.

- Providing the physicians who are specified in cancer and diabetes fields a technology that can assist them in rapidly diagnosing new medical cases with a fraction of time compared to the normal process of diagnosis.
  This means more patients will be treated and more cases will be discovered early.

Once the feasibility study is completed, and we discuss the problems we try to solve, the next step is to define the requirements of the new system. This involves identifying the functional requirements of the new system, such as what tasks it needs to perform, what data it needs to store and retrieve. It also involves identifying the non-functional requirements of the system, such as its performance, security, and scalability.

Functional Requirement :-

- **Login**
    - The system will verify the Email and password that was entered by the user.
- **Sign up**
    - The user will be able to create an account on the application.
- **Patient Medical History**
    - The system will store the user information (blood type, gender, age)
- **Image Classification**
    - The system will apply classification on diagnoses images that the user entered
- **Numeric Classification**
    - The system will apply classification on diagnoses numeric data
- **Result Display**
    - The system will Display the classification results to the user

Non-functional Requirements :-

- **Security**
    - The user accounts should be secured, and only an authorized user can access the information inside the account
- **Scalability**
    - More features could be add, so the system should be adaptable for changes and the increasing in features
- **Availability**
    - When ever the user wants to use the system it should be up and working
- **Performance**
    - The classification should not take longer than 3 seconds to complete, and the navigation through the application should be smooth and fast.

User Story :-

- **As the patient,** i want to have a medical profile in the system so i don't have to enter my personal information every time a use the system
- **As the patient,** I want to be able to take a picture of the disease or upload it from my device as I might not have a picture on my device.
- **As the patient,** I want my personal information to be secured so no one can use my information without my approval.

Mobile features :-

- **Functional**
  - **Register**
    - The user can create an email by typing his name and mobile number.
  - **Login (with email and password**
    - The user can log in using his email that was created at sign up process
  - **Profile.**
    - It consists of the user's profile picture, name and email, blood type, and age.
  - **Skin Cancer and X-ray Functions.**
    - Take photo
    - upload photo
    - Run Model in photo
    - show result
  - **receive diabetes data.**
  - **Display diabetes results.**
- **Non-functional**
  - **dark mode and light mode.**

Web Features :-

- **Sign up**
  - The user can create Account by typing name, mobile number, e-mail, password, gender, birthdate
- **Login (email and password)**
  - The user can log in using his account that was created at the sign-up process.
- **User profiles**
  - It consists of the user's name, email, blood type, and age.
- **Specialized Medical Fields**
  - (x-ray, Skin cancer, diabetes, Brain Tumor)

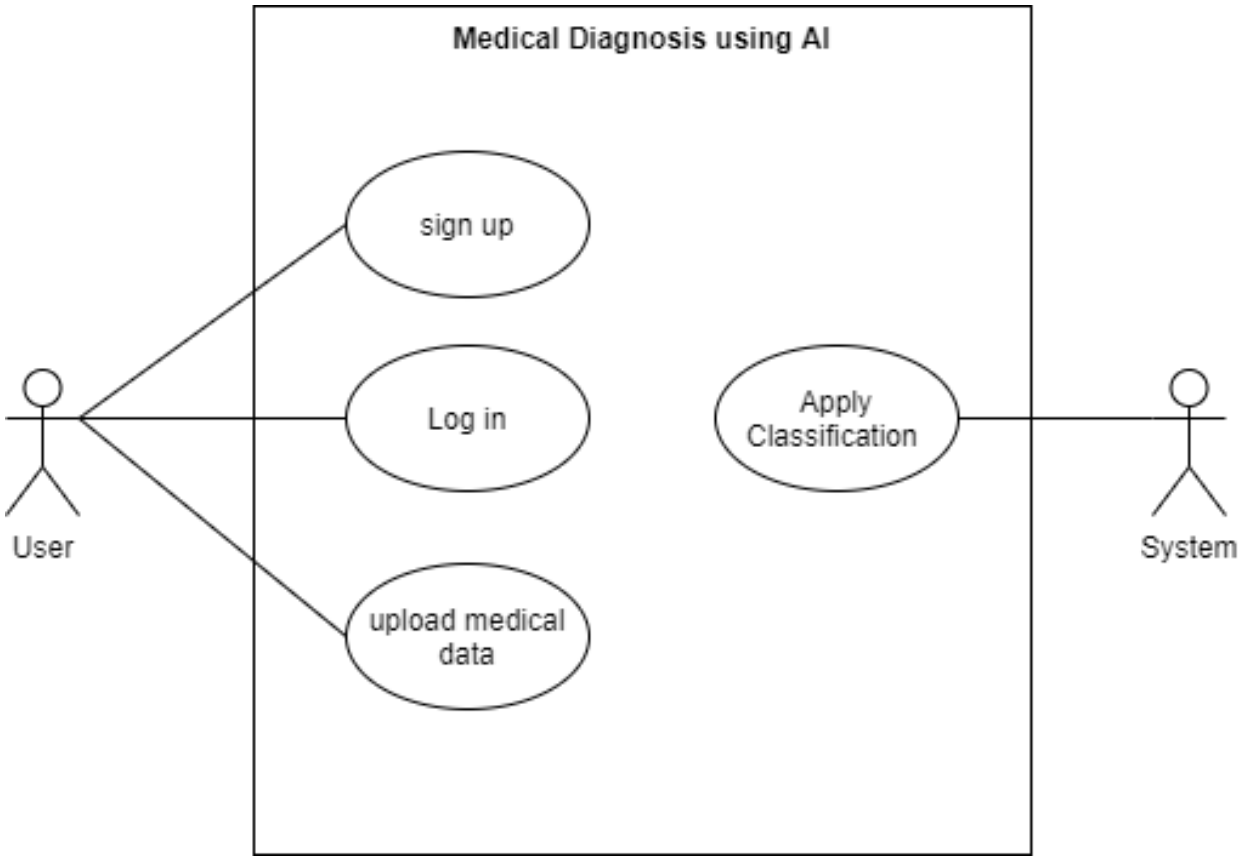
The planning and analysis phase is critical to the success of the system analysis and design process. It sets the foundation for the development of the system and helps to ensure that the system meets the needs of the users and the organization.

2.3.2 Phase 2: Design

The next step is to design the user interface of the system. The user interface is the front-end of the system and is the part that the users interact with. The user interface should be designed to be intuitive, user-friendly, and efficient. It will be discussed further in both **chapter 3** and **chapter 4**

The system analysis and design process involves the use of several tools and techniques to gather and analyze information, and to develop and implement the new system. Some of the common tools and techniques used in system analysis and design include:

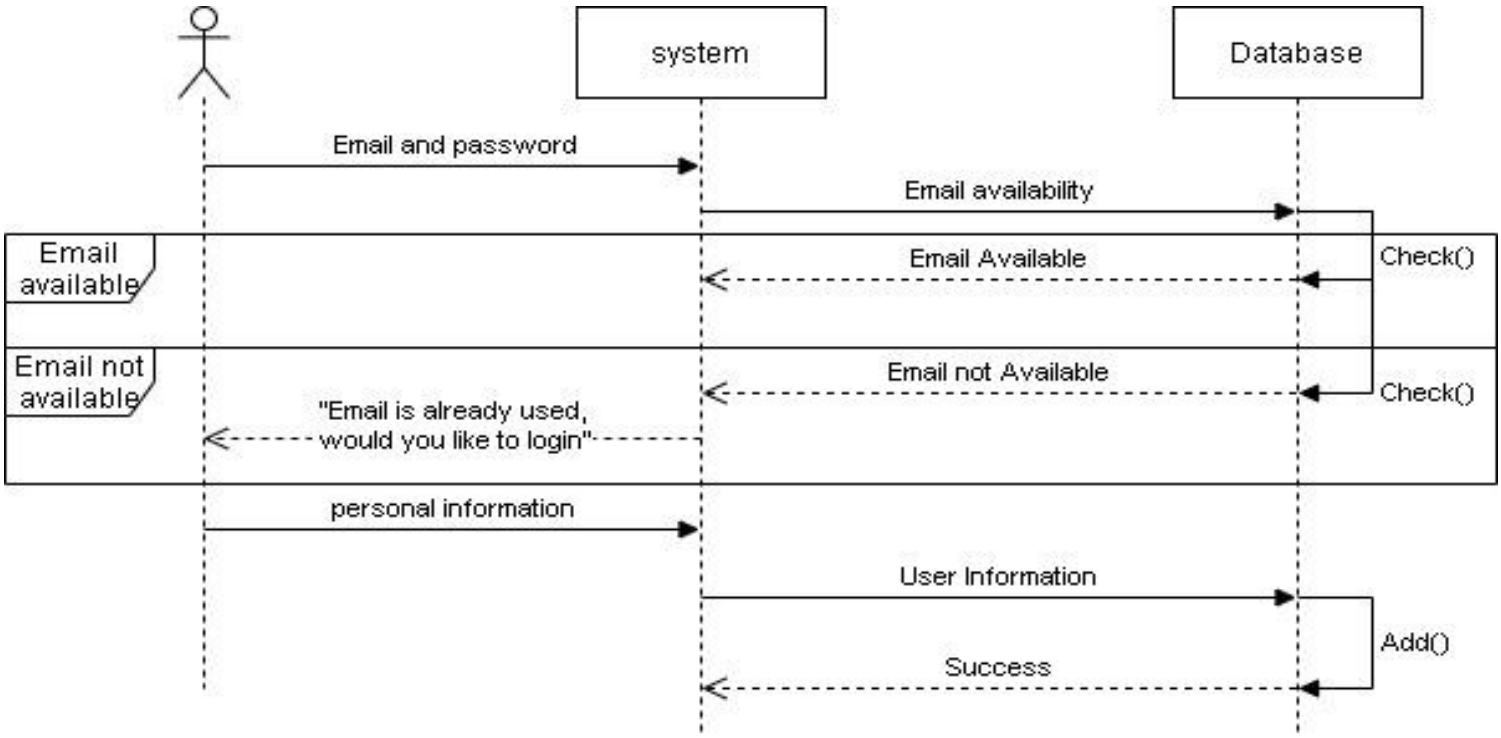
Use Cases: Use cases are scenarios that describe how a user interacts with the system. They are used to identify the functional requirements of the system and to ensure that the system meets the needs of the users.



Sequence diagram: is a type of diagram used to visualize the interactions and messages that occur between objects or components in a system over time. It shows the sequence of events that take place during a particular use case or scenario, including the order in which messages are sent and received by objects or components, and the lifelines of the objects or components involved.

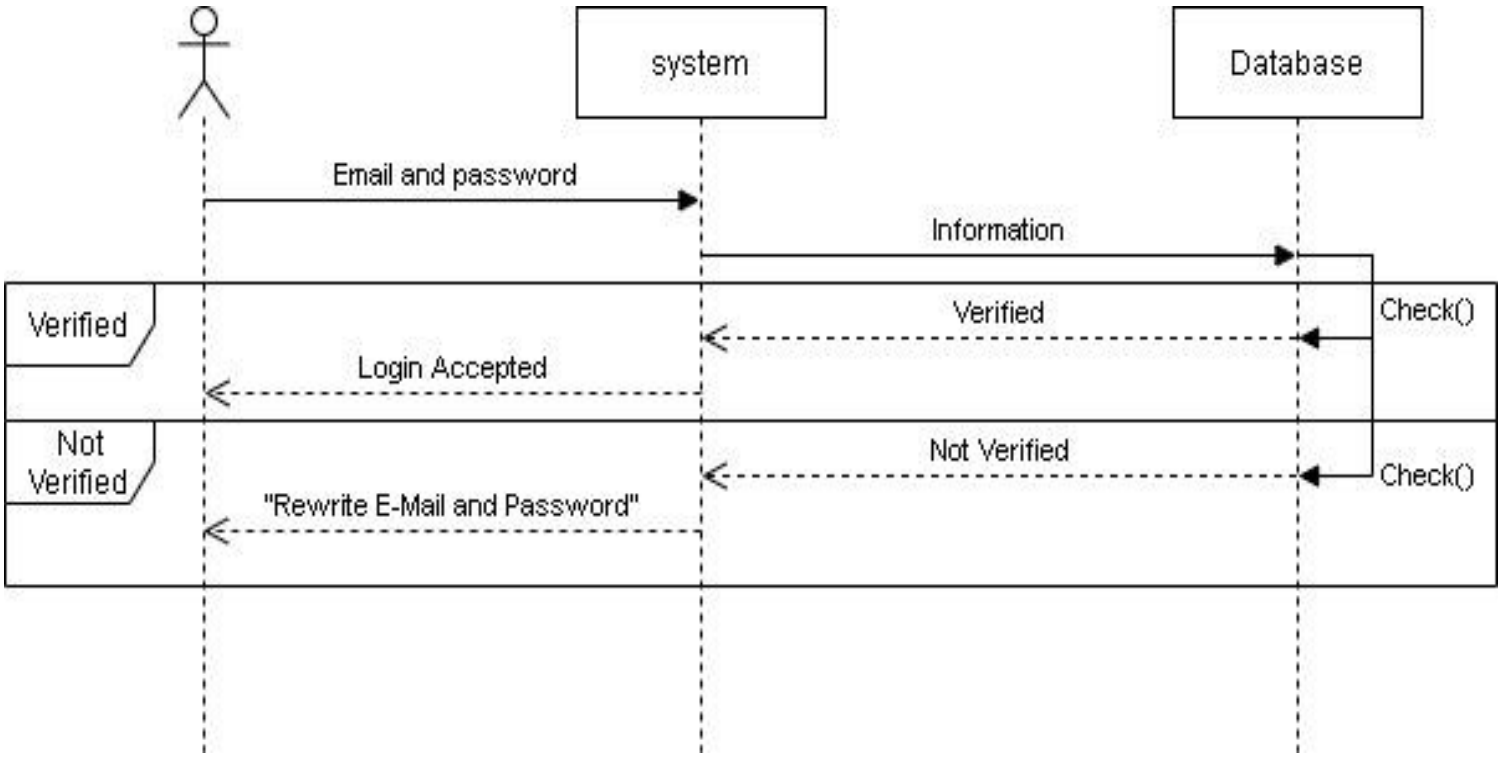
Sign Up

The user is creating a new account in the application, the following sequence diagram demonstrate the interactions and messages that occur between objects or components in a system over time



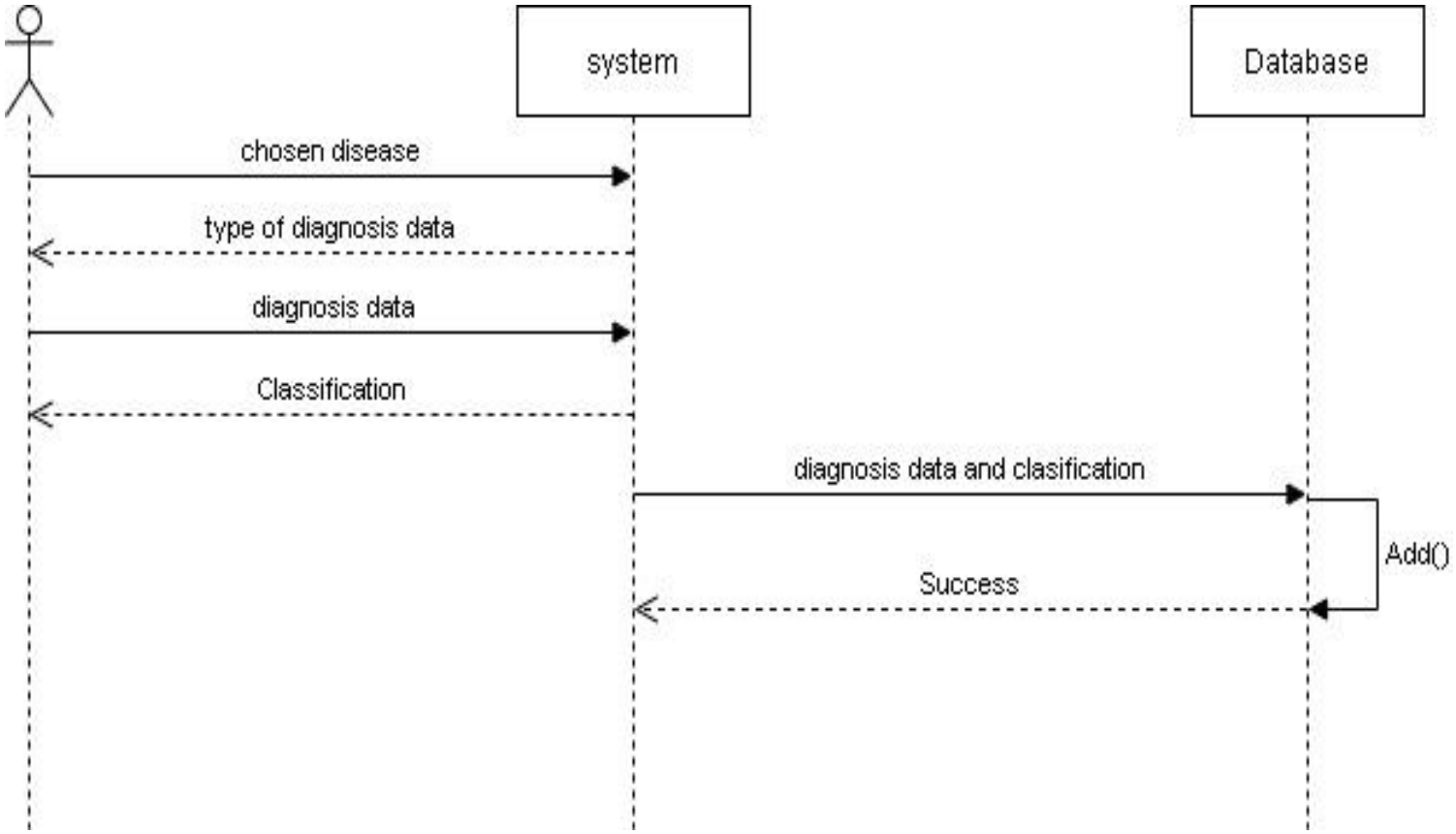
Login

The user is trying to access his account in the application, the following sequence diagram demonstrate the interactions and messages that occur between objects or components in a system over time

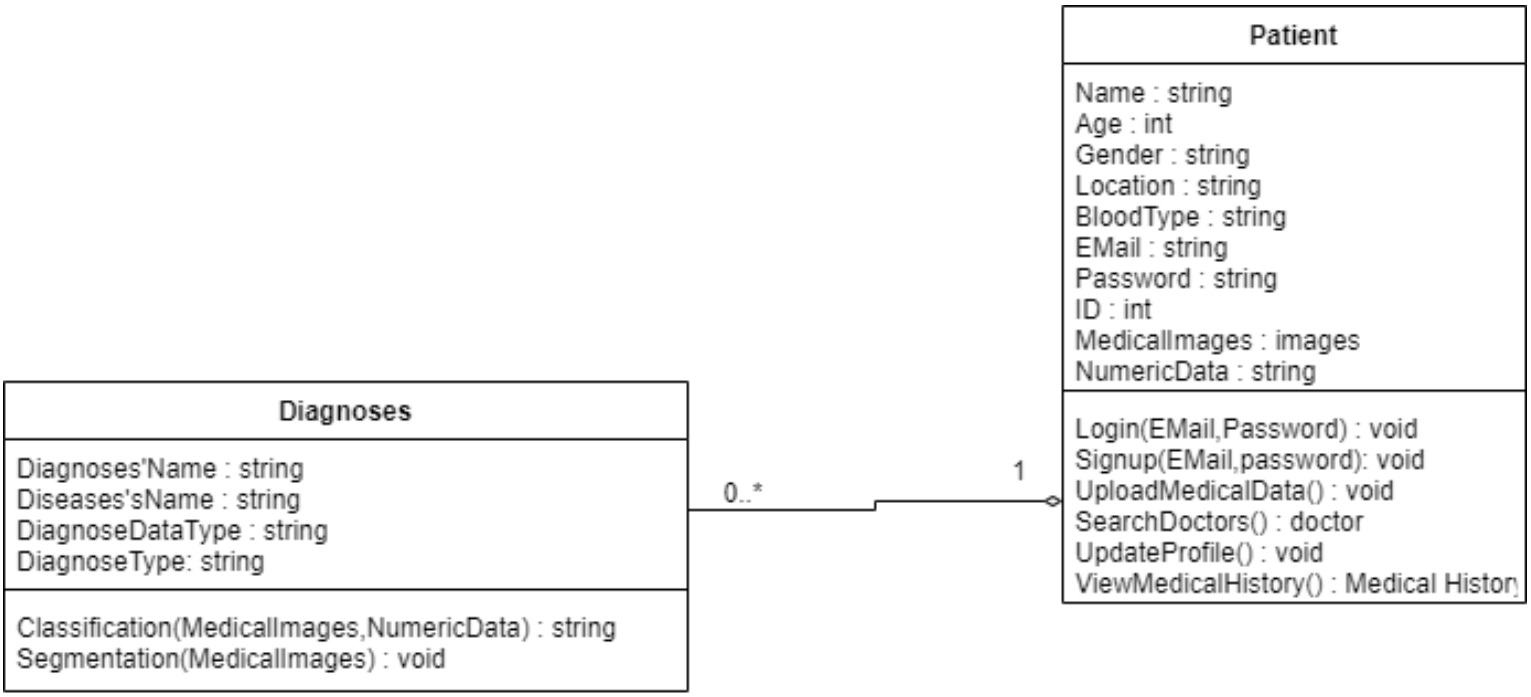


Uploading Medical data

The user is uploading his medical data to the application to apply the diagnoses upon these images, the following sequence diagram demonstrate the interactions and messages that occur between objects or components in a system over time



Class diagram: is a type of diagram used to visualize the structure and relationships of classes in an object-oriented system. It shows the classes, interfaces, attributes, and methods of a system, and how they are connected to each other.



The next step is to design the database of the system. The database is where the system stores and retrieves data. The database design involves identifying the tables, fields, and relationships between them. It also involves developing a data dictionary that defines the data elements used in the system.

Entity-relationship diagrams (ERD): ERDs are diagrams that depict the relationships between entities in a system. They are used to model the data relationships in the system and to identify the tables, fields, and relationships in the database.

2.3.3 Phase 3: Implementation

The implementation phase is the third step of the system analysis and design process. This phase involves developing the system based on the design developed in the previous phase.

This phase is all about developing the software code that will be used to implement the system. This involves writing the code, testing it, and debugging it to ensure that it works as expected.

The implementation phase is critical to the success of the system analysis and design process. It ensures that the system is developed and deployed in a controlled and structured manner, and that it meets the requirements identified in the planning and analysis phase.

Phase 4: Maintenance

The maintenance phase is the final step in the system analysis and design process. This phase involves maintaining and supporting the system after it has been deployed. The goal of the maintenance phase is to ensure that the system continues to meet the needs of the users and the organization.

The maintenance phase includes several activities, such as monitoring the system's performance, identifying and resolving issues, and making updates and enhancements to the system. The maintenance phase also involves providing technical support to the users, such as troubleshooting issues and answering questions.

The maintenance phase is critical to the success of the system analysis and design process. It ensures that the system remains effective and efficient over time, and that it continues to meet the changing needs of the organization.

Chapter 3

Mobile Application

3.1 Introduction to Flutter

Flutter is an open-source mobile application development framework created by Google. It allows developers to build high-performance, cross-platform mobile applications for Android and iOS using a single codebase. Flutter uses the Dart programming language, which was also developed by Google, and provides a rich set of pre-built widgets and tools for building beautiful, responsive UIs.

One of the key features of Flutter is its "hot reload" functionality, which allows developers to see changes to their code reflected immediately in the app without having to recompile. This makes the development process faster and more efficient, enabling developers to iterate quickly and see the results of their changes in real-time.

Another key feature of Flutter is its focus on performance. Flutter apps are compiled ahead of time to native code, which means they can run at near-native speeds. Additionally, Flutter uses a reactive programming model, which allows the UI to be updated in real-time as data changes, without requiring a full re-render of the app.

Flutter also provides a comprehensive set of tools for testing and debugging, including support for unit testing, integration testing, and widget testing. This makes it easy to catch bugs and ensure that your app is functioning as expected.

In summary, Flutter is a powerful mobile development framework that enables developers to build high-performance, cross-platform mobile applications using a single codebase. Its key features include hot reload, a rich set of pre-built widgets, support for animations and transitions, a focus on performance, and comprehensive testing and debugging tools.

3.2 About the application:

Used Packages

1- http package:

The http package in Flutter is a built-in package that provides a way to make HTTP requests and handle responses from an API or server. It allows developers to send HTTP requests such as GET, POST, PUT, DELETE, etc. and receive responses in the form of JSON, XML, HTML, or other formats.

To use the http package in your Flutter app, you need to add it to your pubspec.yaml file and then import it into your Dart code. Here is an example of how to use the http package to make a POST request:

In the above example, the http.post() method is used to make a POST request to the URL.

2-BLoC Package

BLoC (Business Logic Component) is a state management pattern in Flutter that separates the business logic of an application from its UI. With BLoC, the UI sends events to the BLoC, which then processes the events and emits new states. The UI subscribes to the BLoC and updates itself whenever a new state is emitted.

The BLoC pattern consists of three main components:

- **Events**: These are the inputs to the BLoC. Examples of events can be a button click, a text field change, or a network response.

- **BLoC**: This is the main component of the pattern that processes the events and emits new states. The BLoC can be thought of as a black box that takes in events and produces states.

```dart
@override
Future<String> uploadBrainTumor(File? img) async
{
  final request = http.MultipartRequest(
      "POST", Uri.parse(ApiConstance.postBrainTumor));
  final header = {"Content_type": "multipart/form-data"};
  request.files.add(http.MultipartFile(
      'file', img!.readAsBytes().asStream(), img.lengthSync(),
      filename: img.path
          .split('/')
          .last));
  request.headers.addAll(header);
  final myRequest = await request.send();
  http.Response res = await http.Response.fromStream(myRequest);
  print(img.path);
  if (myRequest.statusCode == 200) {
    final resJson = jsonDecode(res.body);
    print("response here: $resJson");
    return resJson['result'];
  } else {
    throw UnimplementedError();
  }
}
```

3-States: These are the outputs of the BLoC. States represent the current state of the application and can be used to update the UI.

To implement BLoC in Flutter, developers can use a combination of StreamController and StreamBuilder. The StreamController is used to manage the events and states, while the StreamBuilder is used to rebuild the UI whenever a new state is emitted.
Here's an example of how to implement BLoC in Flutter:

```dart
abstract class AppState{}

class InitialState extends AppState{}

class ChangeModeState extends AppState{}

class SuccessPick extends  AppState{}

class LoadingPick extends  AppState{}

class ErrorPick extends  AppState{}

class SuccessBrainTumorUpload extends  AppState{}

class LoadingBrainTumorUpload  extends  AppState{}

class ErrorBrainTumorUpload  extends  AppState{}

class SuccessChestUpload extends  AppState{}

class LoadingChestUpload  extends  AppState{}

class ErrorChestUpload  extends  AppState{}

class SuccessSkinCancerUpload extends  AppState{}
```

```dart
Future pickSkinCancerImage() async {
  emit(LoadingPick());
  PickedFile? pickedFile = await picker.getImage(
    source: ImageSource.gallery,
  );
  skinImg = File(pickedFile!.path);
  emit(SuccessPick());
}
Future pickChestImage() async {
  emit(LoadingPick());
  PickedFile? pickedFile = await picker.getImage(
    source: ImageSource.gallery,
  );
  chestImg = File(pickedFile!.path);
  emit(SuccessPick());
}
Future pickBrainImage() async {
  emit(LoadingPick());
  PickedFile? pickedFile = await picker.getImage(
    source: ImageSource.gallery,
  );
  brainImg = File(pickedFile!.path);
  emit(SuccessPick());
}
```

4- Architecture pattern

We used clean architecture.

Clean Architecture is an architectural pattern that emphasizes separation of concerns and promotes a modular design approach for software development. This pattern was introduced by Robert C. Martin and is often used in modern software development to improve code quality, maintainability, and testability.

In a Clean Architecture, the codebase is divided into different layers, where each layer has its own responsibilities and is independent of the other layers. The three main layers of Clean Architecture are:

- Presentation Layer: This layer is responsible for the UI and user interactions. It includes the widgets, pages, and screens that users see and interact with.

- Domain Layer: This layer contains the business logic and rules of the application. It is independent of any specific UI framework or external services and represents the core of the application.

- Data Layer: This layer is responsible for managing the data and communication with external systems such as APIs and databases.

In Flutter, Clean Architecture can be implemented using various packages and design patterns such as BLoC, Provider, and MVC.

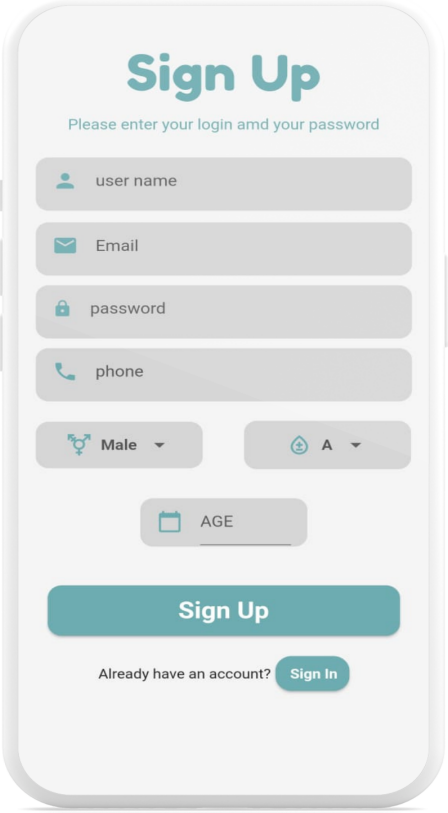
3.3 Application screens

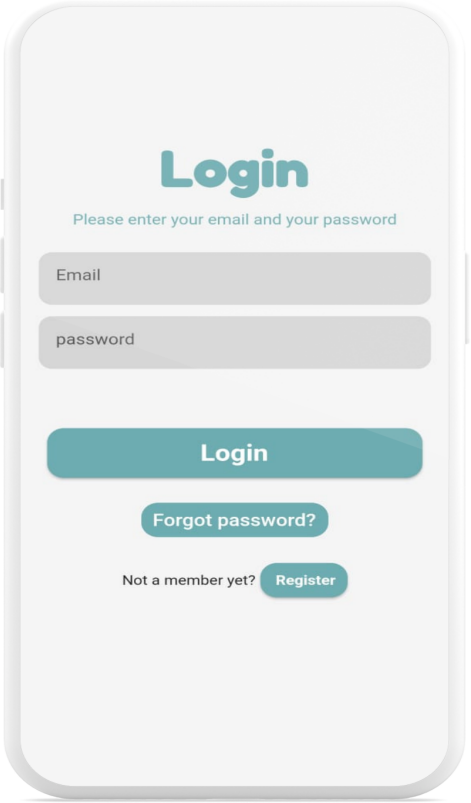
1- Sign up screen

The user can create an account in the application through the sign-up screen by filling in this information, then this data will be transferred to the profile screen.

2-Login screen

The user can log in to the application by entering the email and password that he registered with the sign-up screen.



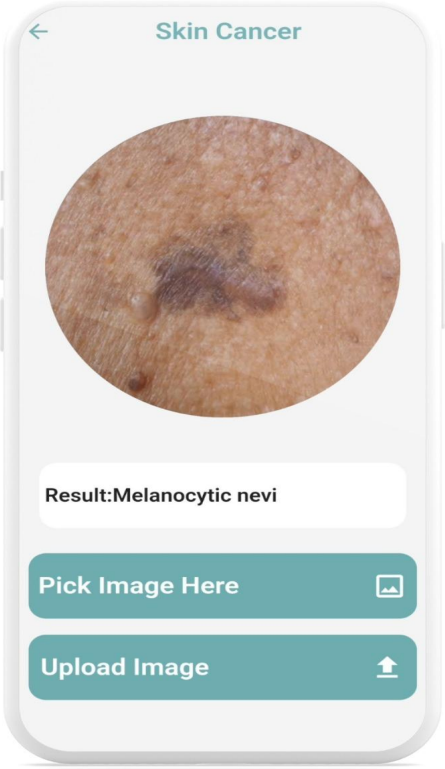


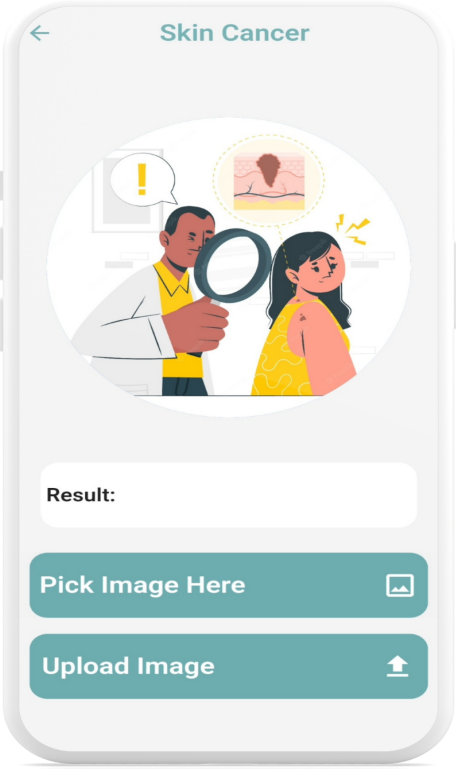
4-Diseases screens

- Diabetes screen
- Skin cancer screen
- brain tumor screen
- chest x-ray screen

For example skin cancer screen:

The user can export the location of the disease, then upload the image to the application, then the application performs the required manipulations on the image to diagnose the disease, and then the diagnosis is shown to the user or patient





Chapter 4

Web Application

4.1 Introduction to Web Application

A web application, also known as a web app, is a software application that is accessed and used through a web browser over a network, typically the internet. It is designed to provide functionality and deliver services to users through a client-server architecture.

Web applications are built using web technologies such as HTML (HyperText Markup Language), CSS (Cascading Style Sheets), and JavaScript. These technologies work together to create the structure, presentation, and interactivity of the application.

Web applications can range from simple and static websites with limited functionality to complex and dynamic platforms with advanced features. They can be used for a wide range of purposes, including e-commerce, social networking, content management, online banking, collaboration tools, and more.

The key components of a web application include:

Front-End: The front-end is the user interface that users interact with directly. It consists of HTML, CSS, and JavaScript code that determines the visual appearance and behavior of the application in the user's browser. Front-end technologies are responsible for rendering content, handling user input, and providing an intuitive user experience.

Back-End: The back-end is the server-side component of the web application that handles data processing, business logic, and database interactions. It is responsible for receiving user requests from the front-end, executing the necessary operations, and generating responses. Back-end technologies include server-side programming languages like Python, Ruby, Java, or Node.js, and frameworks such as Django, Ruby on Rails, Spring, or Express.js.

Database: A web application often requires a database to store and retrieve data. The database is responsible for persistent data storage and management. Commonly used databases include MySQL, PostgreSQL, MongoDB, or SQLite. The back-end interacts with the database to store and retrieve data as needed.

Networking: Web applications rely on networking protocols to communicate between the client-side and server-side components. The HTTP (Hypertext Transfer Protocol) is commonly used for communication between the browser and the server. This enables the exchange of requests and responses, allowing the web application to function.

Web applications offer several advantages, such as cross-platform compatibility, ease of maintenance, and the ability to access them from anywhere with an internet connection. They are widely used in various industries and have become an integral part of our online experiences, providing convenience, accessibility, and powerful functionality to users.

FrontEnd

4.2.1 About the application:

Used Packages & Frameworks

1-Bootstrap framework:

Bootstrap is a free and open-source CSS framework. It is the most popular front-end framework in the world. Bootstrap provides a consistent, responsive, and easy-to-use design system for websites and web applications.

The Bootstrap package includes the following:

- A CSS framework with over 100 pre-built components, including buttons, forms, tables, and navigation menus.
- A JavaScript library with over 30 plugins, including modals, tooltips, and popovers.
- A Sass compiler that allows you to customize Bootstrap to your own needs.

Here are some of the benefits of using the Bootstrap package:

- Responsive design: Bootstrap is designed to work on any device, from desktops to smartphones.
- Modern design: Bootstrap uses the latest web technologies, such as CSS3 and Sass, to create a modern and stylish design.

2-AOS library:

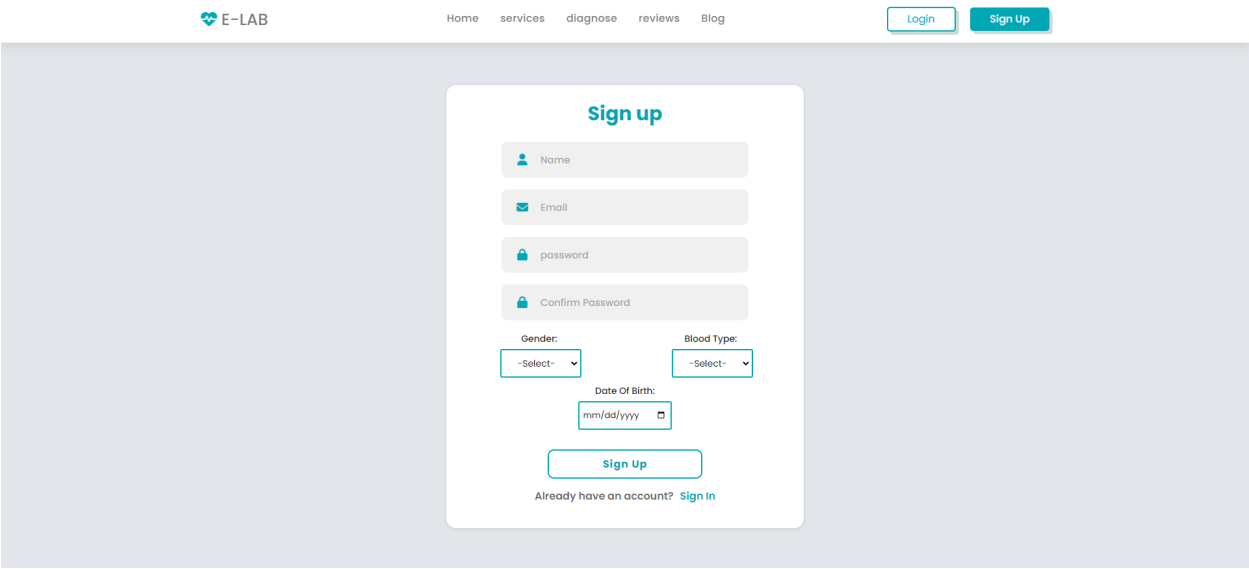
AOS (Animate on Scroll) is a free and open-source CSS library that allows you to animate elements on your page as you scroll. It is a lightweight library that is easy to use and customize.

AOS offers a variety of animation effects, including fade, slide, flip, and zoom. You can also customize the animation speed, delay, and easing.

4.2.2 Application screens

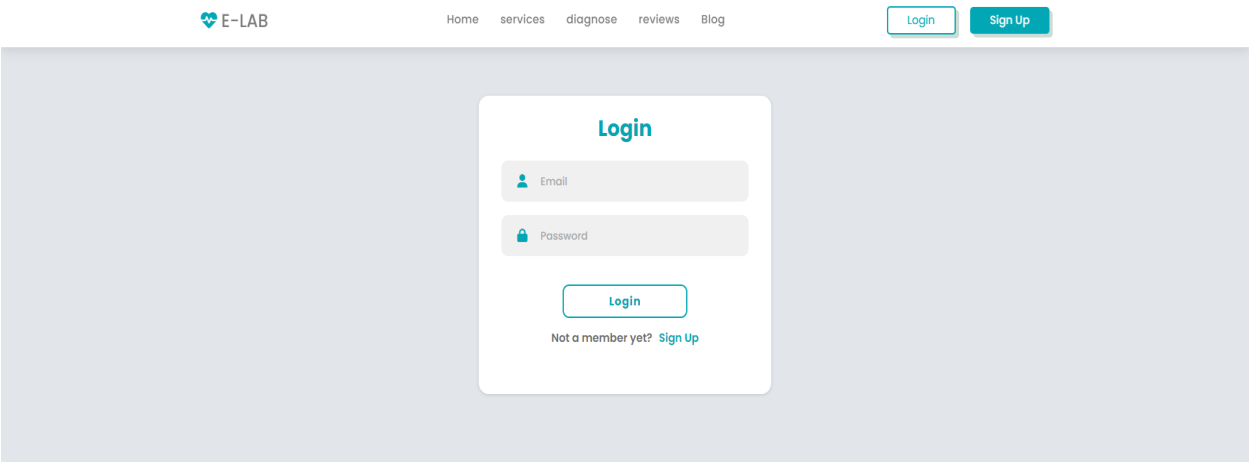
1- Sign up screen

The user can create an account in the application through the sign-up screen by filling in this information, then this data will be transferred to the profile screen.



2-Login screen

The user can log in to the application by entering the email and password that he registered with the sign-up screen.

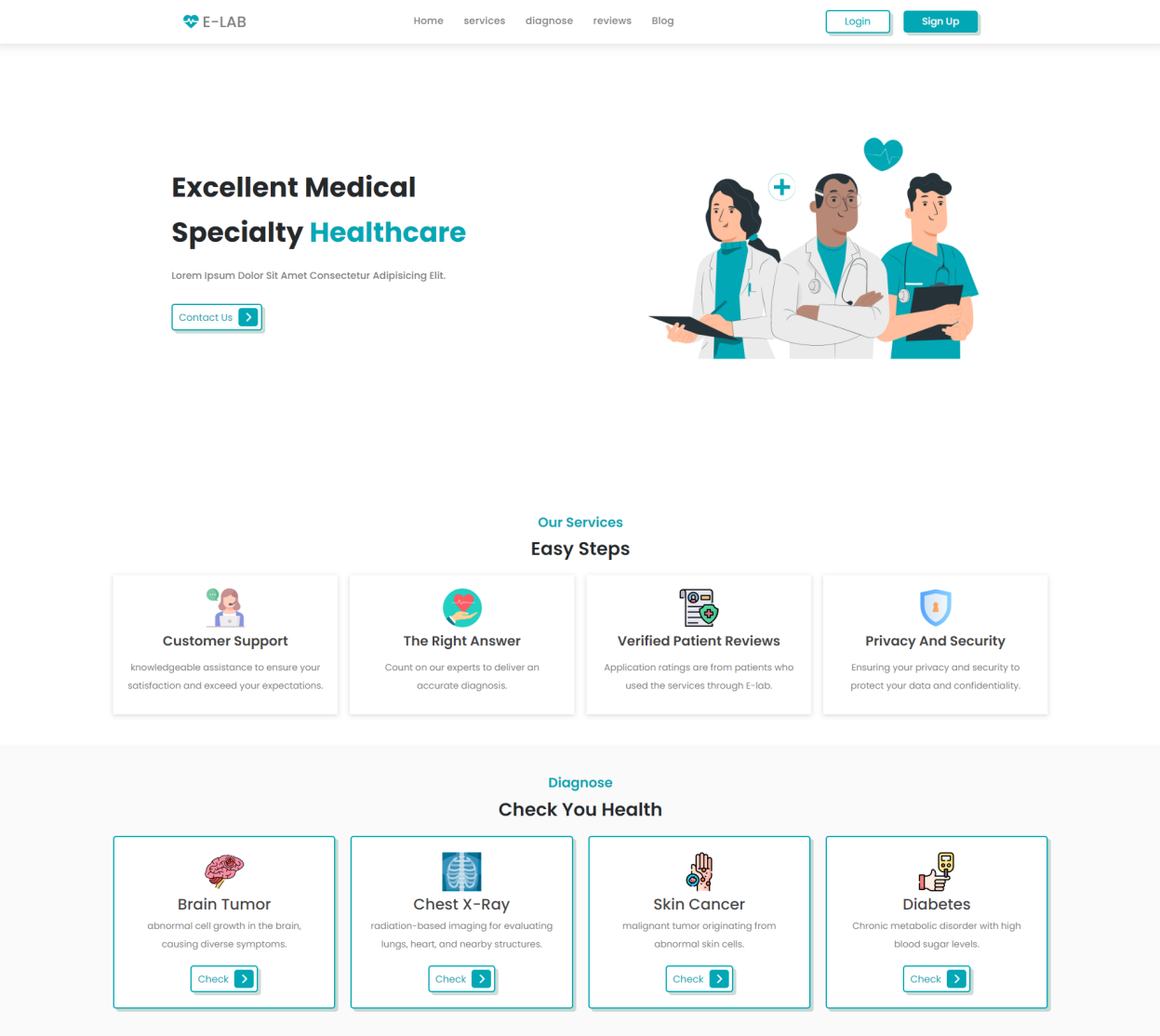


3-Home page

navigation bar we found quick links on the website and login and signup button and logout of application.

A four partition of Home page:

- First part is the services and features of the website.
- Second part is diagnoses, we found four diagnoses and explain what is the disease and button check to review the disease
- Third part is reviews from patients who used E-lab services.
- Fourth part is blog that contain popular articles about diseases.

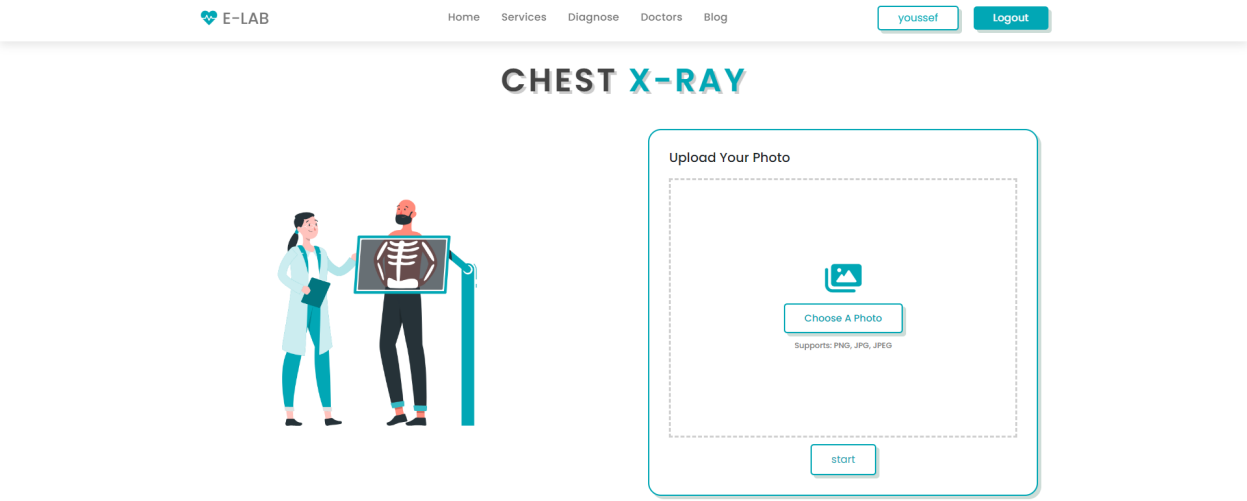


4-Diseases screens

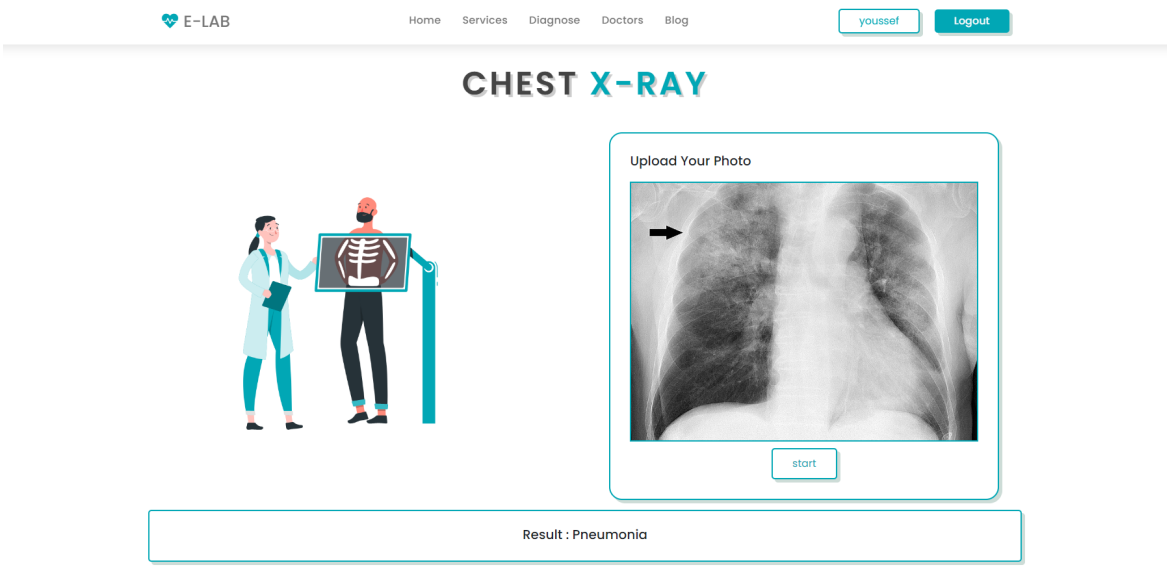
- Diabetes screen
- Skin cancer screen
- brain tumor screen
- chest x-ray screen

For example chest x-ray screen:

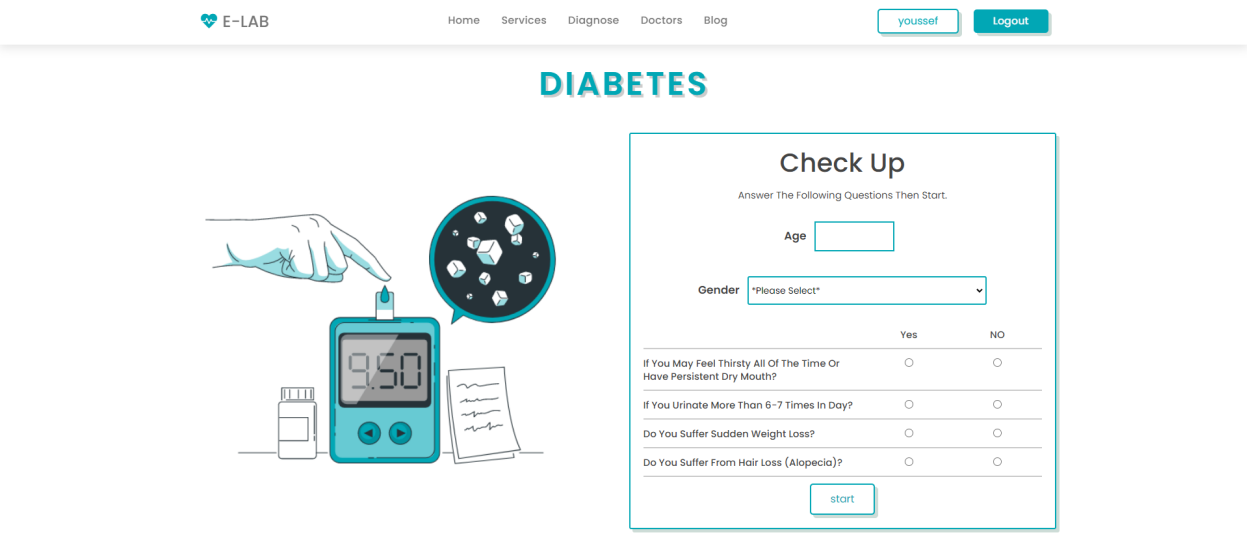
upload the image of the infected part to the application, then the application performs the required manipulations on the image to diagnose the disease, and then the application gives the user or patient the result.



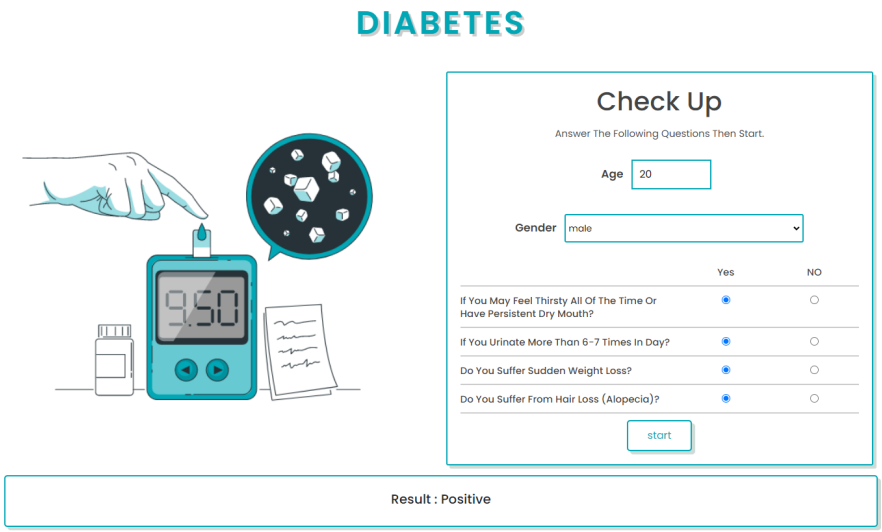
After the upload and diagnoses:



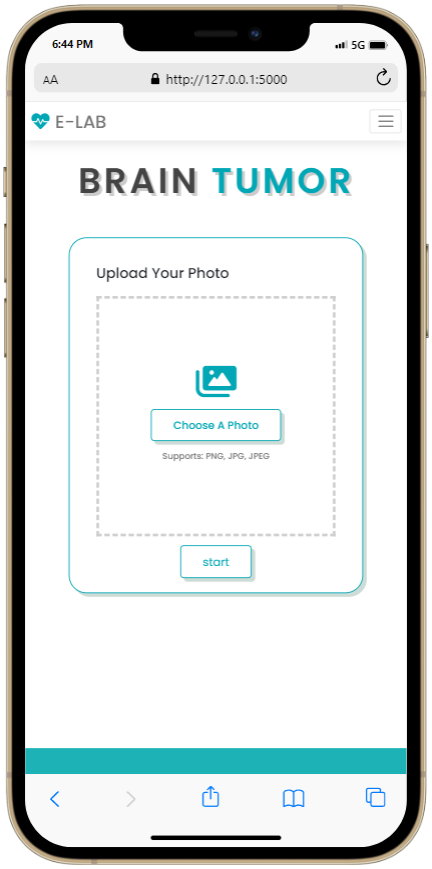
Next webpage is diabetes is form contain a some question and answer with yes or no and in the bottom form found a button start to predict and get the result

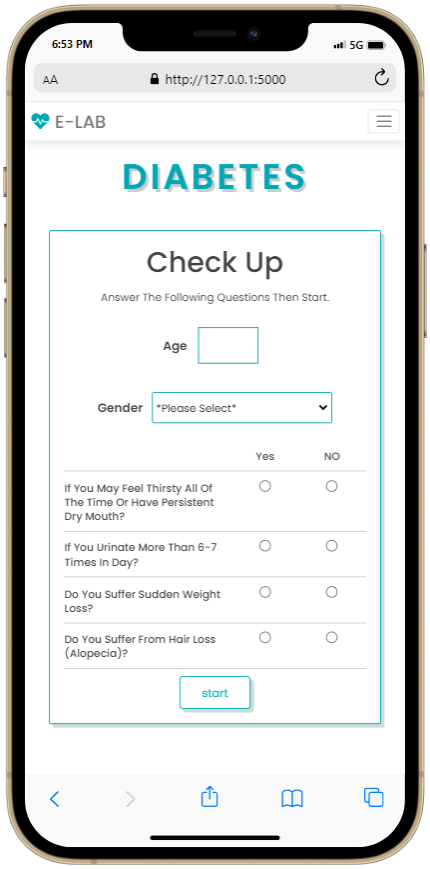


After the upload and diagnoses:



5-Responsive design (Mobile view)





Backend

4.3.1 What is an API?

API stands for Application Programming Interface; APIs are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. For example, the weather bureau's software system contains daily weather data. The weather app on your phone "talks" to this system via APIs and shows you daily weather updates on your phone.

API integrations are software components that automatically update data between clients and servers. Some examples of API integrations are when automatic data sync to the cloud from your phone image gallery, or the time and date automatically sync on your laptop when you travel to another time zone. Enterprises can also use them to efficiently automate many system functions.

4.3.2 How do APIs work?

API architecture is usually explained in terms of client and server. The application sending the request is called the client, and the application sending the response is called the server. So in the weather example, the bureau's weather database is the server, and the mobile app is the client.

There are four different ways that APIs can work depending on when and why they were created.

1. SOAP APIs
2. RPC APIs
3. Websocket APIs
4. REST APIs

4.3.3 What are REST APIs?

REST stands for Representational State Transfer. REST defines a set of functions like GET, PUT, DELETE, etc. that clients can use to access server data. Clients and servers exchange data using HTTP.

The main feature of REST API is statelessness. Statelessness means that servers do not save client data between requests. Client requests to the server are similar to URLs you type in your browser to visit a website. The response from the server is plain data, without the typical graphical rendering of a web page.

4.3.4 REST APIs offer four main benefits:

1. **Integration**: APIs are used to integrate new applications with existing software systems. This increases development speed because each functionality doesn't have to be written from scratch. You can use APIs to leverage existing code.

2. **Innovation**: Entire industries can change with the arrival of a new app. Businesses need to respond quickly and support the rapid deployment of innovative services. They can do this by making changes at the API level without having to re-write the whole code.

3. **Expansion**: APIs present a unique opportunity for businesses to meet their clients' needs across different platforms. For example, maps API allows map information integration via websites, Android,iOS, etc. Any business can give similar access to their internal databases by using free or paid APIs.

4. **Ease of maintenance:** The API acts as a gateway between two systems. Each system is obliged to make internal changes so that the API is not impacted. This way, any future code changes by one party do not impact the other party.

Now, let me how to create an API for a machine learning model:

1. **Choose a framework:** Flask and Django are popular Python web frameworks for building APIs. FastAPI is a newer framework that is gaining popularity due to its performance, simplicity, and automatic documentation features.
2. **Create an endpoint:** An endpoint is a URL that accepts requests from users. You can use Flask's @app.route() decorator or Django's urlpatterns to define endpoints. FastAPI uses Python type annotations to define endpoints and their input and output types
3. **Load the machine learning model:** You can load the trained machine learning model into your API code using Python's pickle module or a more specialized library like joblib or tensorflow.
4. **Prepare the input data:** Preprocessing and data validation may be necessary to ensure that input data passed to the model is in the correct format and within acceptable ranges. For example, you may need to convert strings to numeric values or scale data to match the range of values the model was trained on
5. **Pass the input data to the model:** Once the input data has been prepared, it can be passed to the machine learning model using its predict() or transform() method. Depending on the model, the input data may need to be converted to a specific data type or shape.
6. **Return the output:** The output of the machine learning model can be returned to the user through the API. It may be necessary to convert the output from the model to a more user-friendly format, such as JSON.
7. **Deploy the API:** You can deploy your API on a server using a cloud service like AWS, Google Cloud, or Railway. You may need to configure the server to handle incoming traffic and secure the API using HTTPS
8. **Test the API:** You can test your API using tools like Postman or curl to send requests to the endpoints and verify that the expected output is returned

4.3.5 What is FastApi?

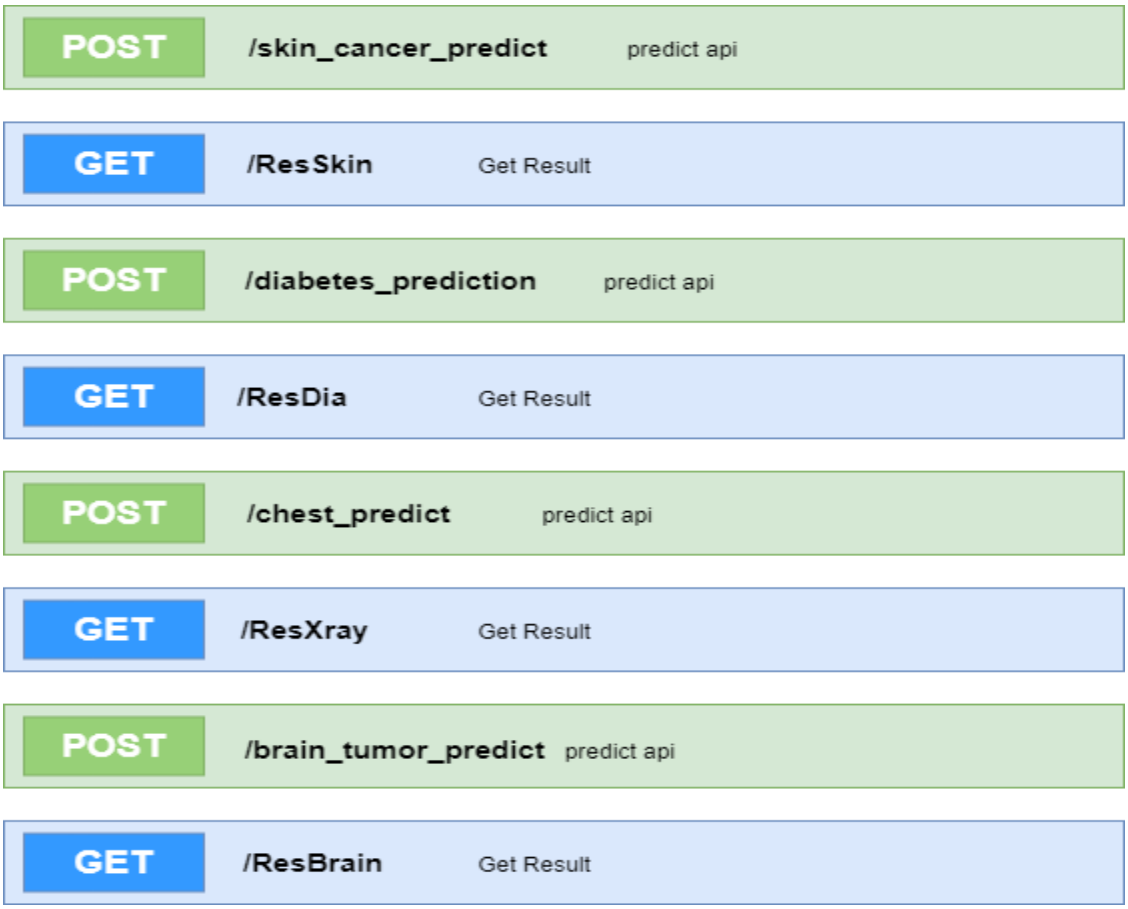
FastAPI is a modern, fast (hence the name) web framework for building APIs with Python. It is built on top of the Starlette framework and relies heavily on Python's type annotations to provide fast and efficient validation, serialization, and documentation of API requests and responses.

Main features of FastAPI

- Fast performance: FastAPI is built with high-performance libraries such as Pydantic and Uvicorn to achieve incredible speeds and handle high traffic loads.
- Simple and easy-to-use: FastAPI comes with a simple and intuitive API that is easy to learn and use.
- Easy validation and serialization: FastAPI uses Python's type hints to automatically validate API requests and responses, as well as to automatically serialize and deserialize data to and from JSON, eliminating boilerplate code.
- Built-in documentation: FastAPI automatically generates interactive API documentation, allowing developers to quickly and easily explore and test APIs.
- ASGI support: FastAPI is built on top of ASGI (Asynchronous Server Gateway Interface), allowing it to leverage the power of asynchronous programming for even faster performance.
- Dependency injection: FastAPI makes use of Python's dependency injection system to automatically inject dependencies into API endpoints and other components, making it easy to build complex applications.

Step1: create endpoints In this project, made four diseases so created four endpoints

Define a function that will handle the incoming requests from users. Use the **@app.post()** decorator to define a POST endpoint that accepts data from the user. Then use the **@app.get()** decorator to define GET endpoint to return result to JSON form



Step2: Load the machine learning model

Load four models using the extension for these models.hdf5 and .sav.Use **tf.keras.models.load_model()** to read the files models to make predict

Step3 & Step4: Prepare the input data

Preprocessing and data validation may be necessary to ensure that input data passed to the model is in the correct format and within acceptable ranges. For example, you may need to convert strings to numeric values or scale data to match the range of values the model was trained on

In a skin cancer model made if the image is not RGB convert it to RGB then resize the image image.resize((28,28)) because the trained machine model is trained with image size is width 28 and height 28 and channel 3. Define the label mapping in the model is 7 diagnosis is defined previous in chapter 3 machine learning

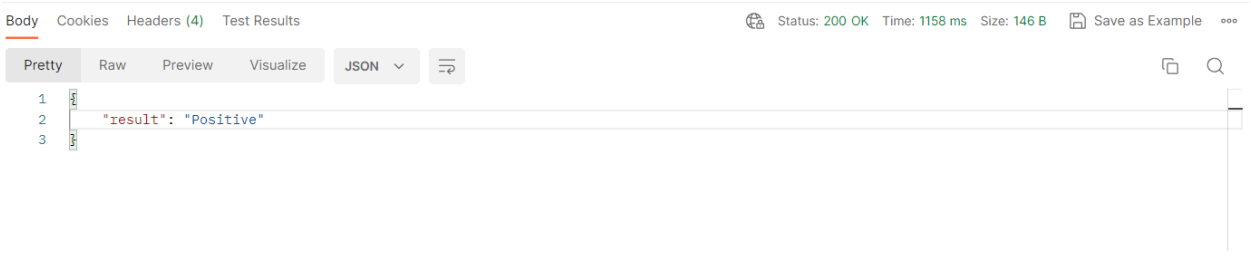
In a brain tumor model made if the image is not RGB convert it to RGB then resize the image image.resize((150, 150)) because the trained machine model is trained with image size is width 150 and height 150 and channel 3. Define the label mapping in the model is 4 diagnosis is defined previous in chapter 3 machine learning

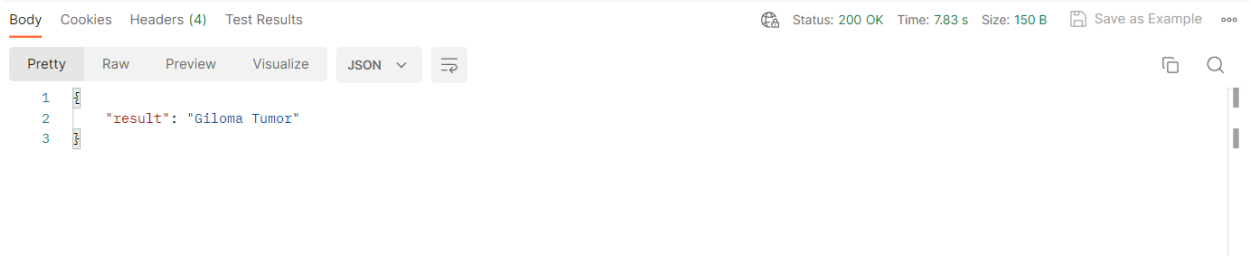
In a chest x-ray model made to convert to grayscale then resize the image image.resize((150, 150)) because the trained machine model is trained with image size is width 150 and height 150 and channel 1. Define the label mapping in the model is 2 diagnosis is defined previous in chapter 3 machine learning

In diabetes model made class to define the features in the model and data type 6 features Polydipsia, Polyuria, Gender, Age, Sudden Weight loss, Alopecia all six features is integer data type then make array to predict the model

Step5: Return the output

The output of the machine learning model can be returned to the user through the API. It may be necessary to convert the output from the model to a more user-friendly format, such as JSON





Step6: Deploy the API

What is a Railway?

Railway is a platform that provides a cloud-based development environment for building and deploying web applications. It allows developers to create, share, and deploy web applications quickly and easily, without having to worry about infrastructure management, server setup, or deployment workflows.

Railway supports a wide range of programming languages, frameworks, and databases, including Node.js, Python, Ruby, PHP, Go, and more. It provides a simple and intuitive web interface for creating and managing applications, and integrates with popular version control systems like GitHub and GitLab for seamless deployment.

With Railway, developers can focus on writing code and building features, rather than worrying about server maintenance or deployment processes. The platform takes care of scaling, security, and other infrastructure-related tasks, making it easy to deploy applications to production with just a few clicks.

In addition to its cloud-based development environment, Railway also provides a number of other features and tools to help developers build and deploy web applications more efficiently. These include:

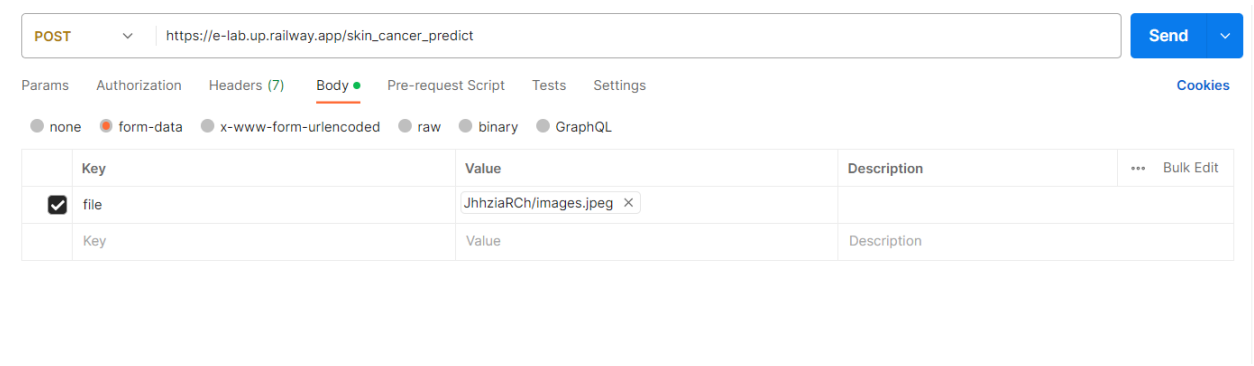
- Automated backups and disaster recovery
- Custom domains and SSL certificates
- Real-time logs and metrics
- Collaboration tools for teams
- Integration with third-party services like Stripe, Twilio, and Sendgrid

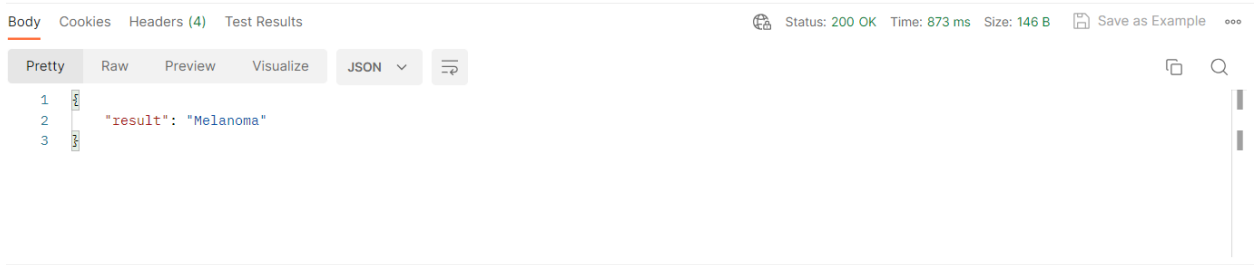
Integrations: Railway integrates with a variety of third-party services, including databases like PostgreSQL and MongoDB, and API services like Stripe and Twilio. Developers can easily connect their applications to these services using Railway's web interface, and manage them from a single dashboard. Project domain is **“https://e-lab.up.railway.app”**

Step7:test the API

Test API using tools like Postman or curl to send requests to the endpoints and verify that the expected output is returned

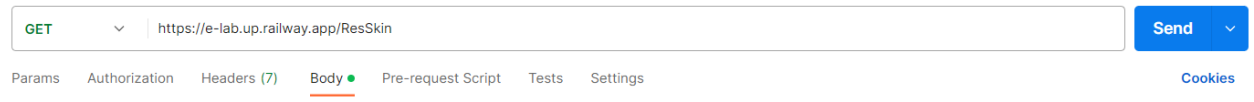
Firstly test POST skin cancer model



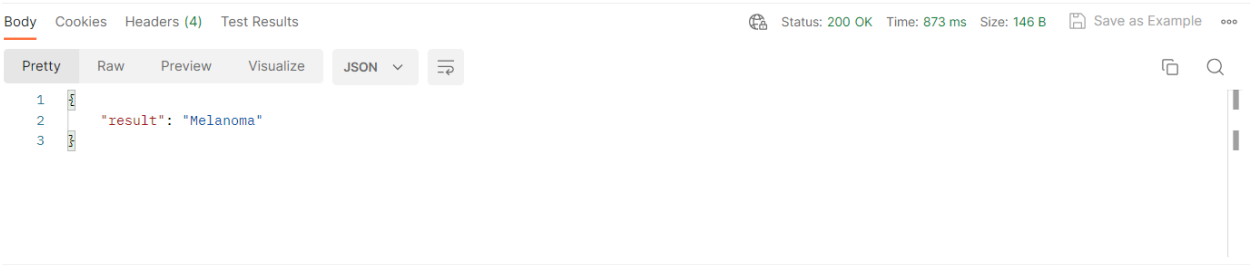


Result

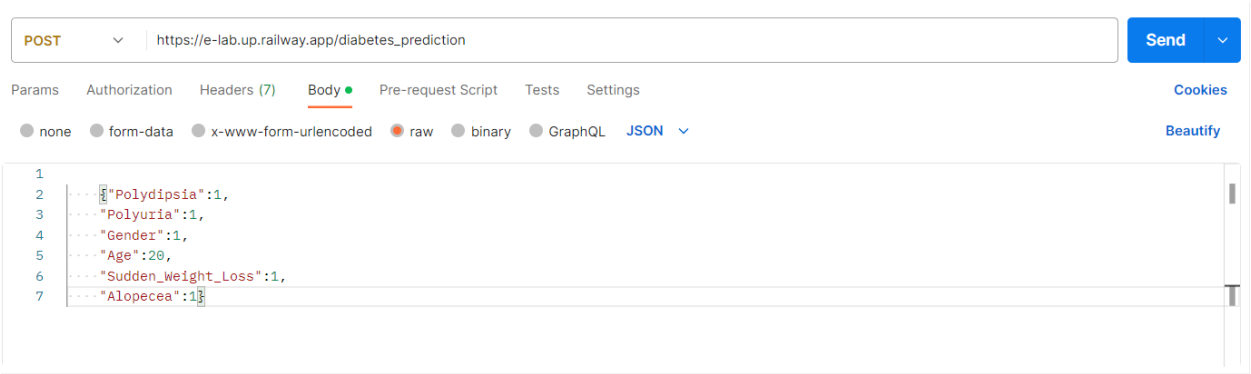
Test GET skin cancer



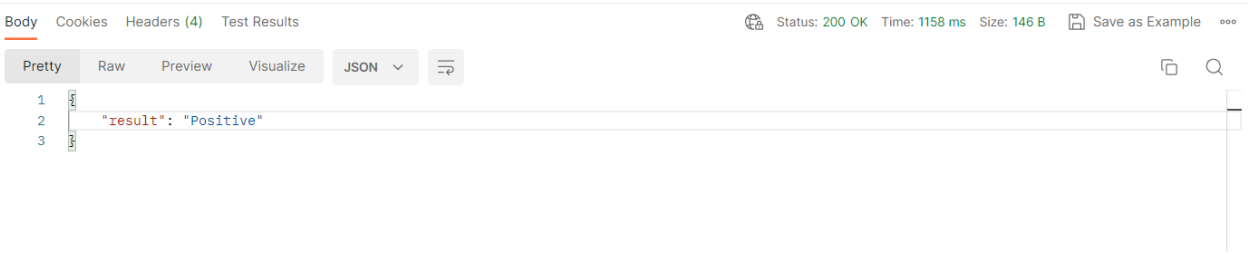
Result



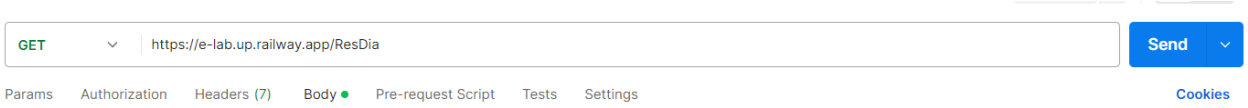
Test POST diabetes model for Positive answer



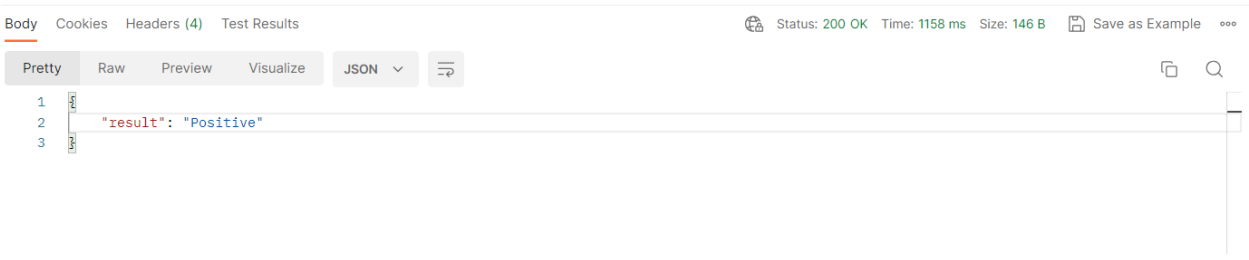
Result



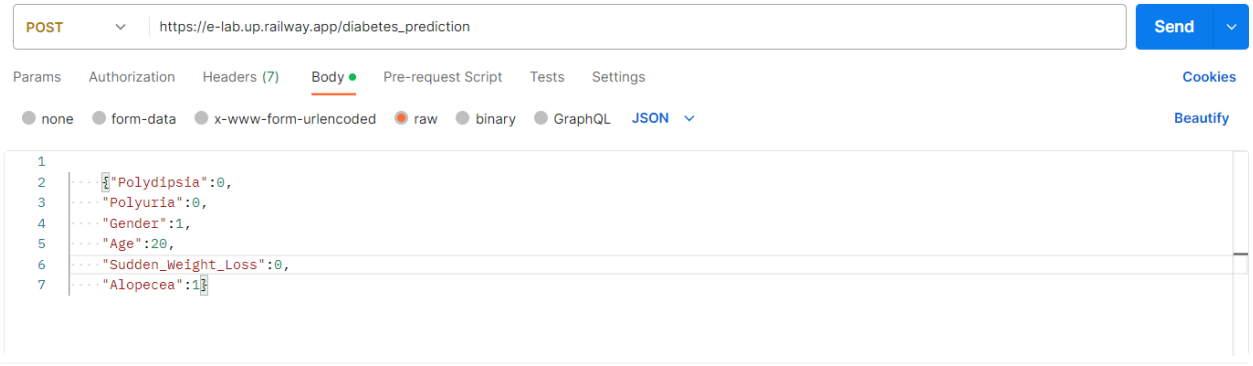
Test GET diabetes model



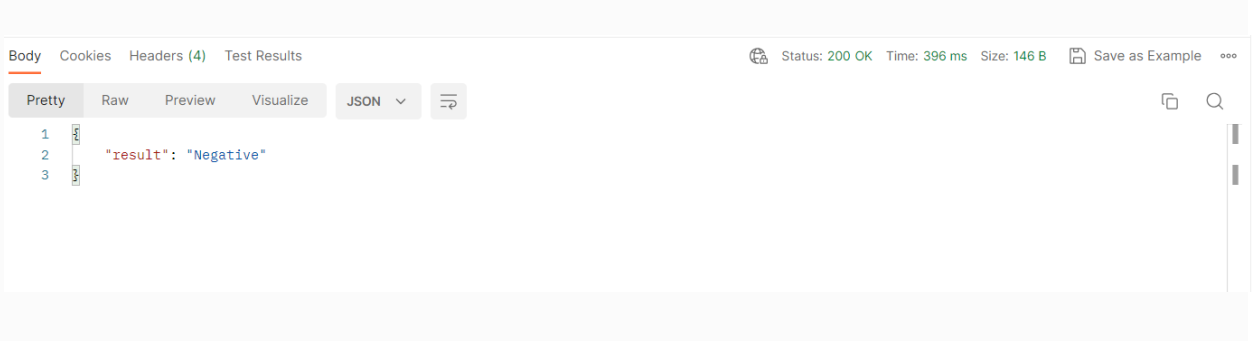
Result



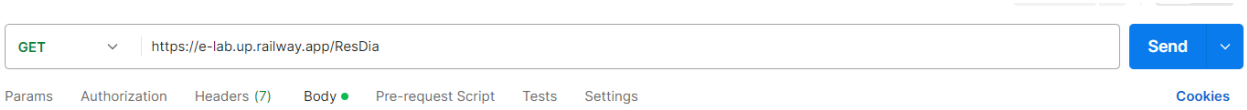
Test POST diabetes model for Negative answer



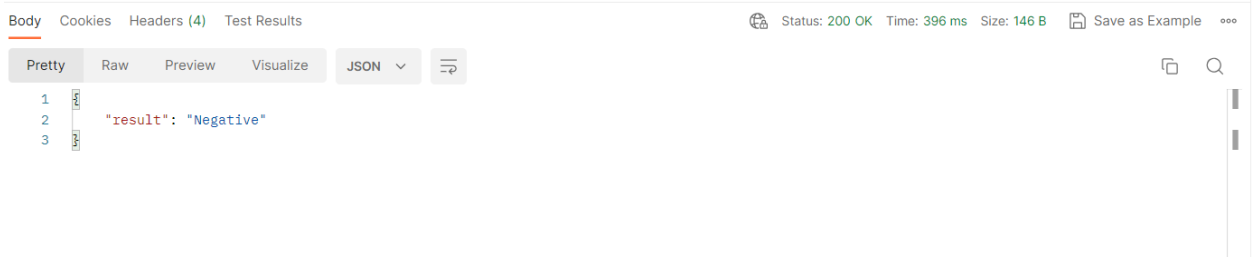
Result



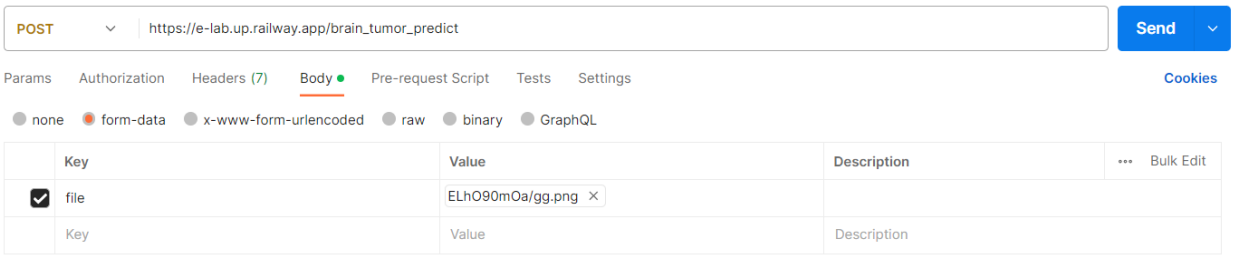
Test GET diabetes model



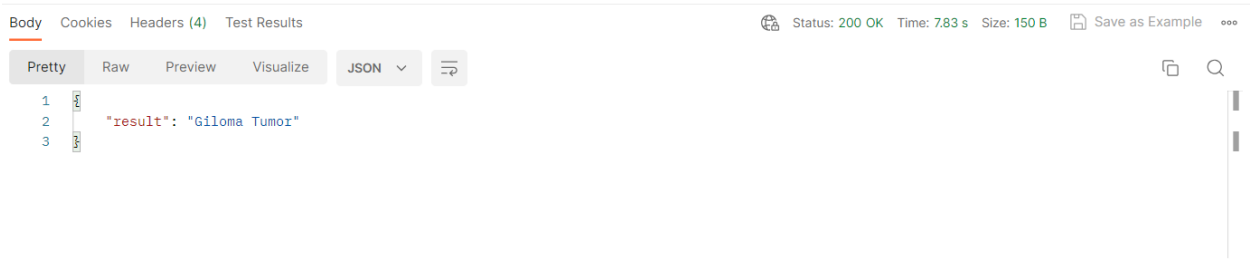
Result



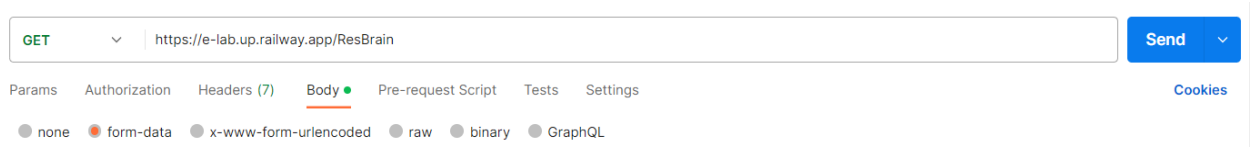
Test POST brain tumor model



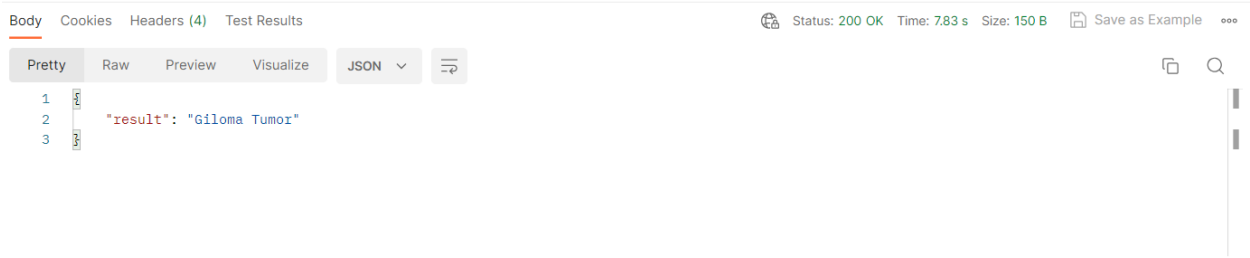
Result



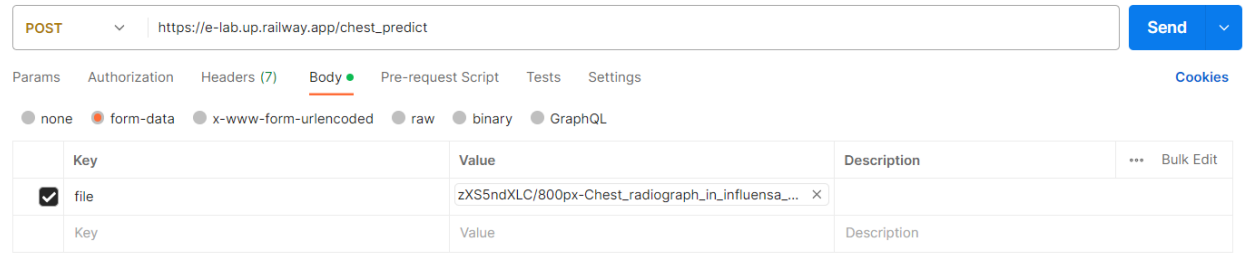
Test GET brain tumor model



Result

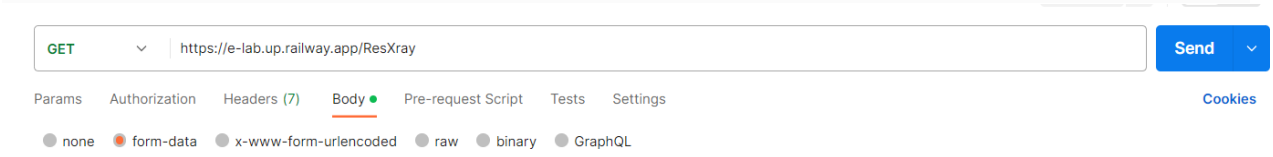


Test POST chest x-ray model

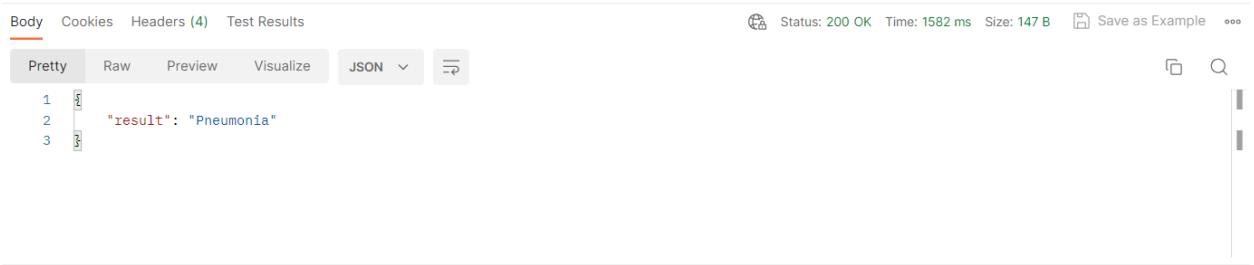


Result

Test GET chest x-ray model



Result



4.3.6 What Is Authentication in Python?

When referring to authentication in Python, we talk about user authentication concerning web applications built with it. Python is actively used in making web applications with many supporting frameworks, including but not limited to Flask, Django, FastAPI, Bottle, and Hug.

Every web application built with Python at one point or another would need to implement user authentication features. This article will cover implementing authentication and proper handling of user identity information using LoginRadius and Flask.

4.3.7 What is flask

Flask is a micro web framework for Python that allows you to quickly build web applications. It is called a "micro" framework because it aims to keep the core of the framework small and simple, while providing extensibility through a variety of plugins and extensions.

Flask was developed by Armin Ronacher in 2010 and has since become one of the most popular web frameworks for Python. It is known for its simplicity, flexibility, and ease of use.

Here are some of the key features of Flask:

1.Routing: Flask allows you to define routes that map to specific functions or methods. This makes it easy to create web pages that respond to specific URLs.

2.Template rendering: Flask provides a simple templating engine called Jinja2 that allows you to generate dynamic HTML pages.

3.Request handling: Flask provides tools for handling incoming requests, such as parsing incoming data (e.g. form data) and accessing query parameters.

4.Response handling: Flask provides tools for generating HTTP responses, such as returning HTML pages, JSON data, or other content types.

5.Extension support: Flask has a large ecosystem of extensions that provide additional functionality, such as database integration, authentication, and more.

Define the Flask in variable app; configure the database link with using SQLite database

4.3.8 Why use SQLite?

SQLite is a popular open-source relational database management system (RDBMS) that is widely used in embedded systems, mobile devices, and small-scale applications. Unlike other RDBMSs like MySQL or PostgreSQL, SQLite does not have a separate server process and can be embedded directly into an application.

SQLite stores data in a single file, which makes it easy to distribute and use in small-scale applications. It supports a subset of SQL (Structured Query Language) and provides a wide range of features for managing data, including:

1. **Data manipulation:** SQLite provides SQL for querying, inserting, updating, and deleting data in tables.

2. **Transactions:** SQLite supports transactions, which allow multiple changes to a database to be treated as a single, atomic operation.

3. **Indexing:** SQLite allows you to create indexes on tables to improve query performance.

4. **Security:** SQLite provides features for securing data, including user authentication and encryption.

5. **Scalability:** SQLite is not designed for large-scale applications and is most suitable for small-scale applications with a limited amount of data.

Make a class to create database columns, make columns first column in an ID and data type is an integer and define ID as a primary key, second column is username and data type is a string (30) with unique variable and third column is a password and data type is a string (30).

Make a class for User Registration form and check if usernames are, exist or not if the username is validate return a message error is “That username already exists. Please choose a different one.”

Make a class for User Login form to enter the username and password if the username and password is not compatible return a message error is “The password is incorrect”

Define the two routes of the pages first route in login page **`@app.route('/login',methods=['POST','GET'])`**, Second route in register page **`@app.route('/signup',methods=['POST','GET'])`**

Chapter 5

Machine Learning Models

5.1 Diabetes disease Machine learning Algorithms

What does diabetes affect the world?

For more than 100 million adults in the United States now living with diabetes or prediabetes, anticipating diabetes side effects early makes treatment 90% successful, so this is

Challenge on project:

Challenge to predict the side effects to detect the side effects and treat the disease in the first stage before it reaches an advanced stage and is difficult to treat. The different steps in this model project.

Firstly loading the essential libraries and then loading and reading the dataset. Then visualizing the dataset using charts, graphs, etc., to understand the data better, then Pre-processing the data using various data cleaning and manipulation techniques, then Identifying the best parameters for the classification models using hyperparameter tuning, then applying multiple machine learning algorithms to train the models and evaluate their performance using different metrics.

5.1.2 Dataset Information

Name of the dataset is *“Early stage diabetes risk prediction dataset”* is collected using direct questionnaires from the patients of Sylhet Diabetes hospital in Sylhet, Bangladesh and approved by a doctor. Data Set Characteristics is multivariate, the number of Instances is 520, and number of attributes is 17.

5.1.3 Attribute Information

1.Age: age of patient is a integer

2.Gender: sex of patient male or female

3.Polyuria: Is a condition where the body urinates more than usual and passes excessive or abnormally large amounts of urine each time you urinate frequent passage of large volume of urine, more than 3 liters per day compared to the normally daily urine output in adults of about 1 to 2 liters (Yes, No).

4.Polydipsia: is the term given to excessive thirst and is one of the initial symptoms of diabetes, it is also usually accompanied by temporary or prolonged dryness of the mouth (Yes, No).

5.Sudden Weight Loss: Is the leading a risk factor for type 2 diabetes, insufficient insulin prevents the body from getting glucose from the blood into the body's cells to use as energy, when this occurs, the body starts burning fat and muscles for energy, Causing a reduction in overall body Weight (Yes, No).

6.Weakness: Might result when the cells do not get enough glucose, diabetes medications, such as insulin or metformin. Help more of this sugar to move into the cells and prevent it from building to harmful levels in the blood (Yes, No).

7.Polyphagia: Is the medical term used to describe excessive hunger or increased appetite and is one of the three main signs of diabetes.

8.Genital thrush: Is a yeast infection (Candida albicans) which tends to affect warm, moist areas of the body such as mouth and certain areas of skin. thrush is more common in people with diabetes as high sugar levels lead to better conditions for the yeast to grow (Yes, No).

9.Visual Blurring: One of the common signs of diabetes mellitus, which refers to the loss of sharpness of vision and the inability to see fine details (Yes, No).

10.Itching: It can be caused by a yeast infection, dry skin or poor circulation. When poor circulation is the cause of itching the itchiest areas may be the lower parts of legs (Yes, No)**.**

11.Irritability: Changes in blood sugar level can affect a person's mood and mental status when blood sugar returns to normal range these symptoms often resolve. Fluctuations in blood glucose can result in rapid mood changes, including low mood and irritability (Yes, No).

12.Delayed Healing: People with uncontrolled diabetes may develop poor circulation. As circulation slows down blood moves more slowly, which makes it more difficult for the body to deliver nutrients to wounds. As a result, the injuries heal slowly or may not heal at all (Yes, No)

13.Partial paresis: paresis involves the weakening of a muscle or group of muscles; it may also be referred to as partial or mild paralysis. Unlike paralysis, people with paresis can still move their muscles. These movements are just weaker than normal (Yes, No).

14.Muscle Stiffness: Lessened ability to move your joints, joint swelling deformities and a "Pains and needles" sensation in the arms or legs. Some musculoskeletal problems are unique to diabetes other also affect people without diabetes (Yes, No).

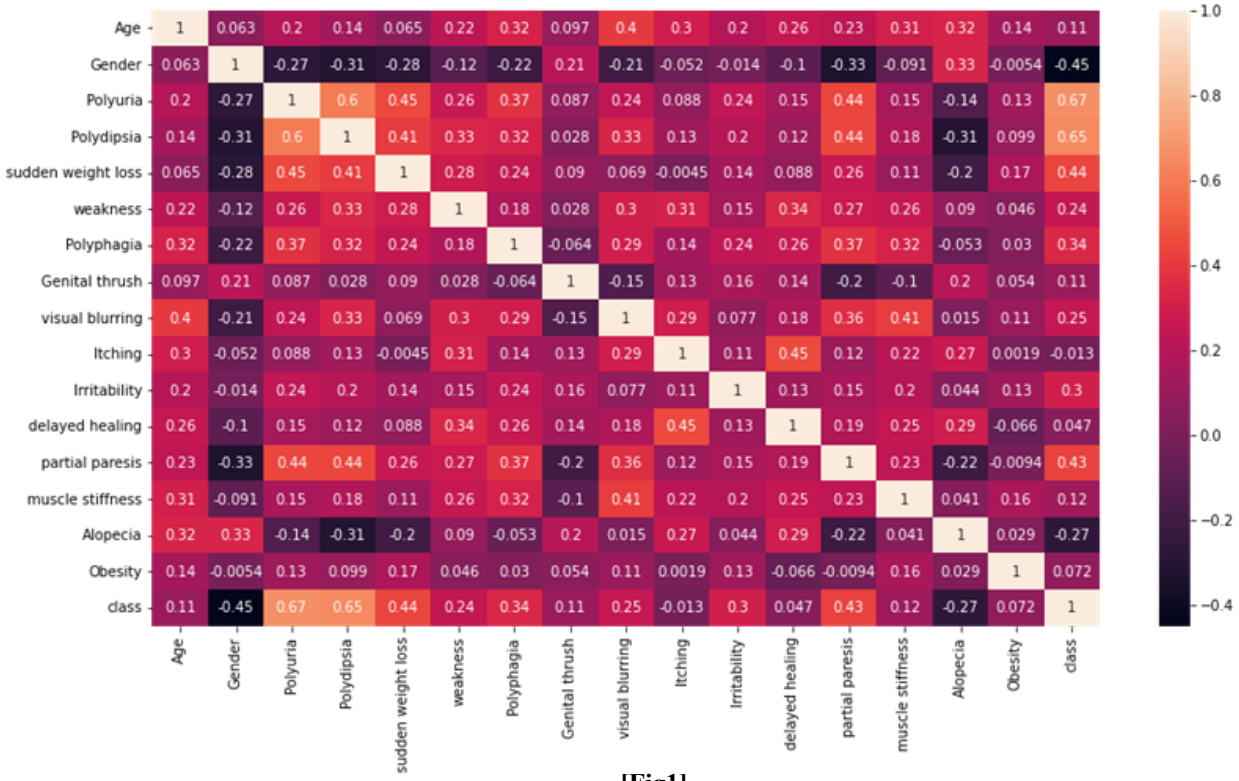
15.Alopecia: The immune system attacks the hair follicles, leading to patches hair loss on the head and other parts of the body (Yes, No)

16.Obesity: Is the leading risk factor for type 2 diabetes. (Yes, No)

17.Class: Positive mean he has a diabetes and Negative mean he has no diabetes

5.1.4 Diabetes Structure/Layout

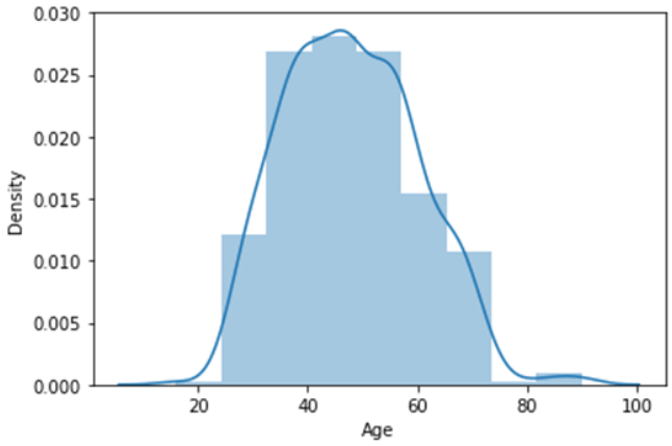
Using library Pandas to read CSV file then check in data. Firstly check the data **info()** to check all attributes and all rows. Then check in the missing value in the data with **function isna().sum()** and find no missing values in data that sound good. So, change all data to numeric using **LabelEncoder()** change all object values to numeric to make correlation to see most attribute useful in model and found The variables Polyuria and Polydipsia are highly Positive correlated, Gender is negative correlated when see in **[Fig1].**



[Fig1]

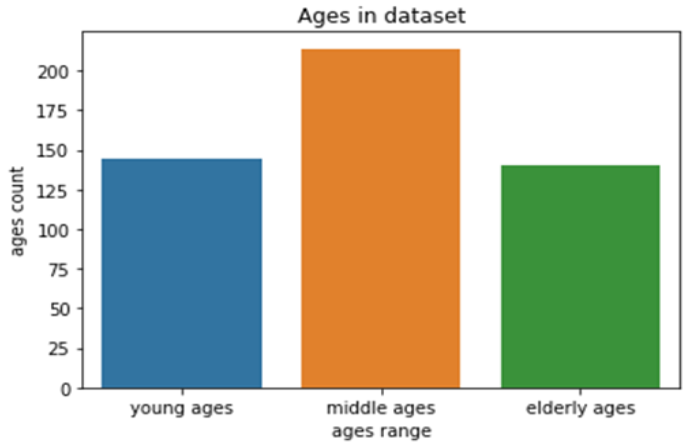
5.1.5 Visualization

Make visualization to more understanding the data and see all data as plots better than to see numeric. Take “**Age**” column and make visualization to see what this column effect on the data, make distribution plot by using library Seborn and using this function in Seaborn distplot() to draw distribution curve and see this column is normal distribution when see in **[Fig 2]** that is good.



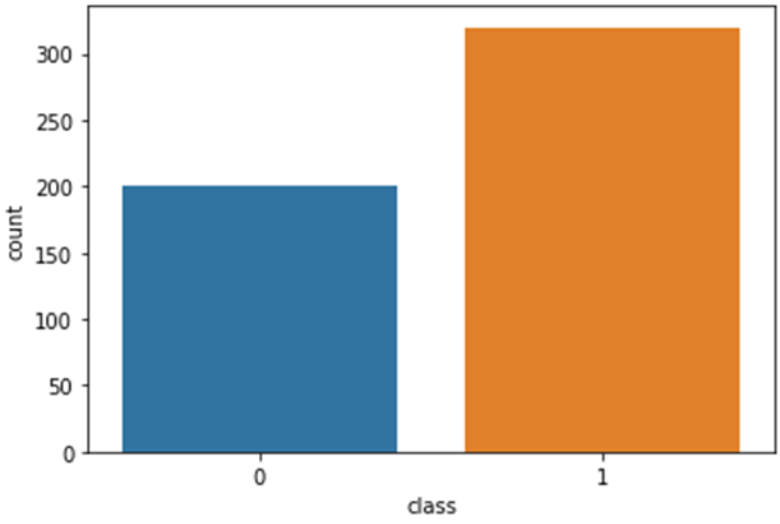
[Fig 2]

I need some information on the “Age” column so I split the "Age" column into three parts. First part called “Young_ages” range of young age is 16 to 40, Second part called “Middle_ages” range of middle age is 40 to 55 and Third part called “Elderly_ages” range of elderly age is 55, I found the “Young_ages” is 144, “Middle_ages” is 214, and “Elderly_ages” is 140 that is mean the middle age is Highest range in column “Age” When see in **[Fig 3]**



[Fig 3]

In the graph **[Fig 4]** to check whether data needs to be imbalanced or not in the column “Class”. We found the data do not need to be imbalanced, the 0 is non-diabetic is 200 with percentage 38.46% and the 1 is diabetic presence is 320 with percentage 61.53%. In this graph use the Seborn library with function **countplot().**



[Fig 4]

5.1.6 Feature selection

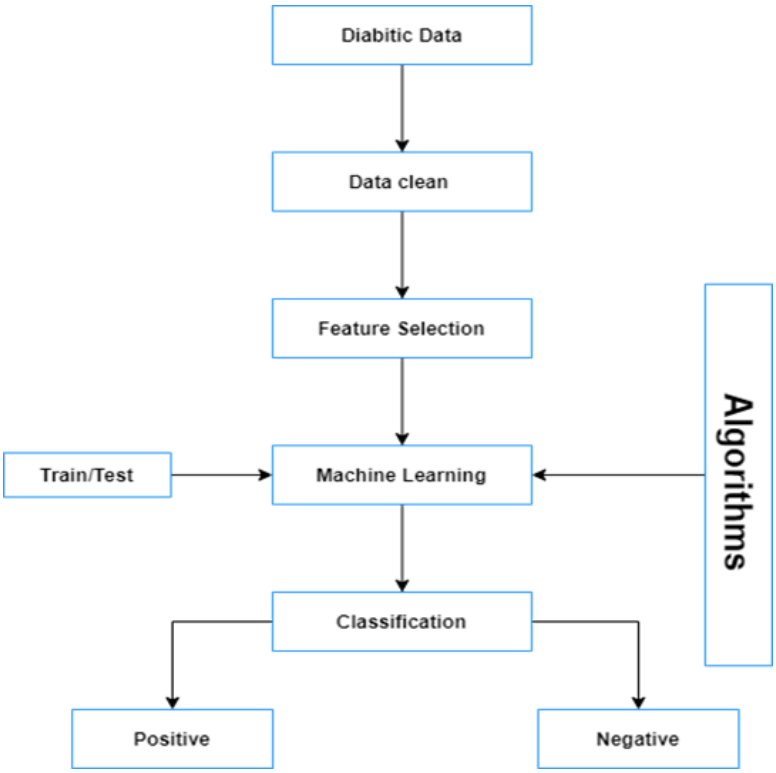
Optimizing the performance of the classification model by feature selecting is an important part. Feature selection, as a data preprocessing strategy, has been proven to be effective and efficient in preparing data (especially high-dimensional data) for various data mining and machine-learning problems. The objectives of feature selection include building simpler and more comprehensible models, improving data mining performance, and preparing clean, understandable data. The feature selection, which has 16 features including: Age, Gender, Polyuria, Polydipsia, Sudden Weight loss, Weakness, Polyphagia, Genital Thrush, Visual blurring, Itching, Irritability, Delayed healing, Partial paresis, muscle stiffness, Alopecia and Obesity, has been applied.

5.1.7 Machine learning models

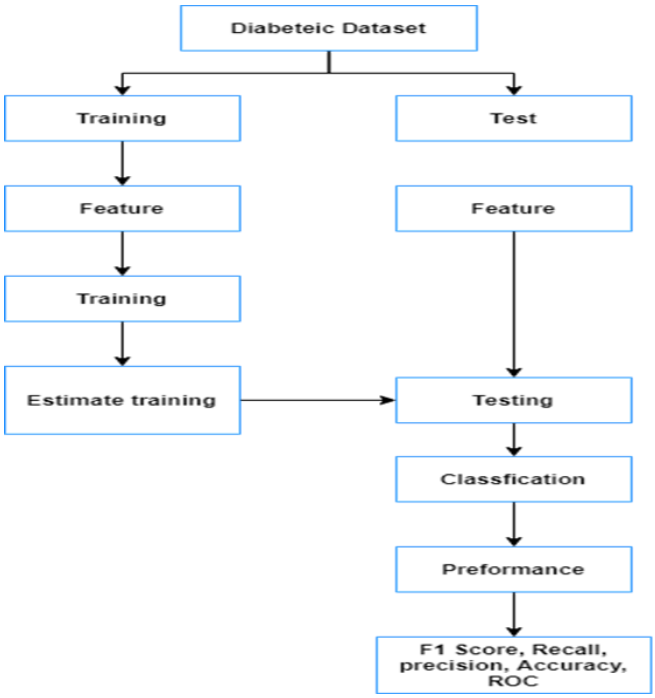
The main objective of the machine learning models is to classify diabetes disease. The overview of the proposed machine learning models has been shown in **[Fig 5].**

The training/test set paradigm of the entire machine learning models has been shown in Fig 6. The first step is to divide the dataset into two sets such as 80% for the training set and 20% for the testing set. The training and test sets are separated. In the second step, the most significant risk factors of diabetes disease have been selected based on PCA feature selection.

We have adopted four classifiers; logistic regression, K-nearest neighbors, support vector machine (SVM) and random forest (RF). The next step is to estimate the training classifier coefficients, and then the test classifiers have been applied to classify the patients into two categories as diabetic vs. control. Finally, the performances of the classifiers are evaluated using five performance parameters, namely accuracy, F1-score, recall, precision and ROC.


Diabitic Data
Data clean
Feature Selection
Algorithms
Train/Test
Machine Learning
Classification
Positive
Negative

[Fig 5]


Diabeteic Dataset
Training
Test
Feature
Feature
Training
Estimate training
Testing
Classfication
Preformance
F1 Score, Recall, precision, Accuracy, ROC

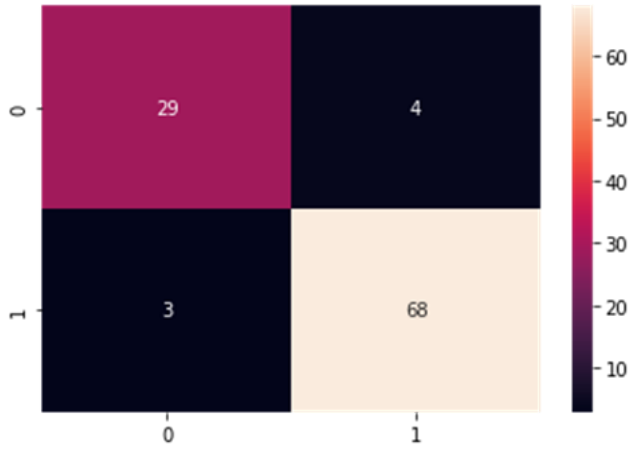
[Fig 6]

5.1.8 Results

In the section, we show the performance of machine learning classification techniques for diabetes classification. For this, we analyze various popular classification techniques that include the logistic regression, K-nearest neighbors, support vector machine (SVM) and random forest (RF). Moreover, four classifiers have been also adapted and compared their performance based on accuracy, F1-score, recall, precision and ROC. The next section represents the related work.

5.1.9 Logistic regression

Logistic Regression is one of the most common classification algorithms. Get training score is 92.06% and test score is 93.27% in **[Fig 7]** draw a plot of TP, TN and FP, FN in a confusion matrix.



[Fig 7]

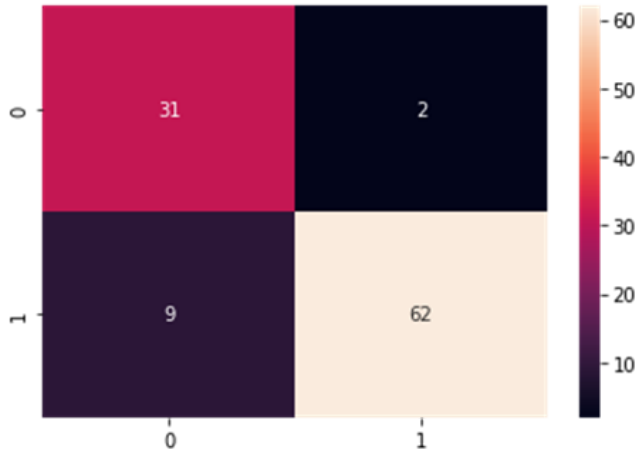
In **[Fig 8]**, show the F1score is 95%, recall is 96% and precision is 94%

[Fig 8]

	precision	recall	f1-score	support
0	0.91	0.88	0.89	33
1	0.94	0.96	0.95	71
accuracy			0.93	104
macro avg	0.93	0.92	0.92	104
weighted avg	0.93	0.93	0.93	104

5.1.10 K-nearest neighbor

The k-NN algorithm is arguably the simplest machine learning algorithm. Get training score is 97.35% and test score is 89.42% in Fig 9 draw a plot of TP, TN and FP, FN in a confusion matrix.



[Fig 9]

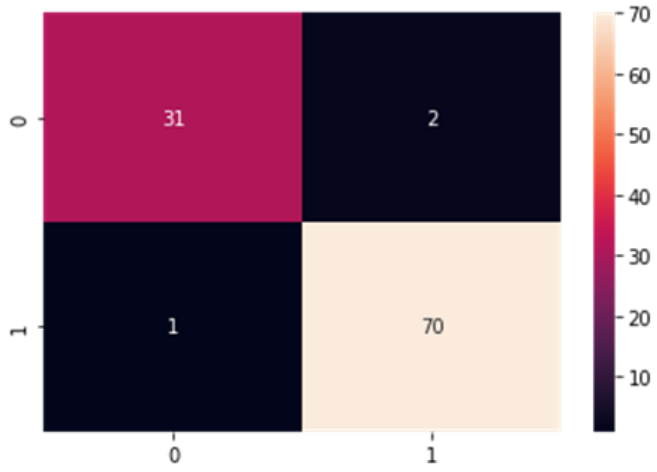
In **[Fig 10],** show the F1score is 92%, recall is 87% and precision 97%

	precision	recall	f1-score	support
0	0.78	0.94	0.85	33
1	0.97	0.87	0.92	71
accuracy			0.89	104
macro avg	0.87	0.91	0.88	104
weighted avg	0.91	0.89	0.90	104

[Fig 10]

5.1.11 Support Vector Machine (SVM)

Is Second best model in the data. Get training score is 99.51% and test score is 97.11% in **[Fig 11]** draw a plot of TP, TN and FP, FN in a confusion matrix.



[Fig 11]

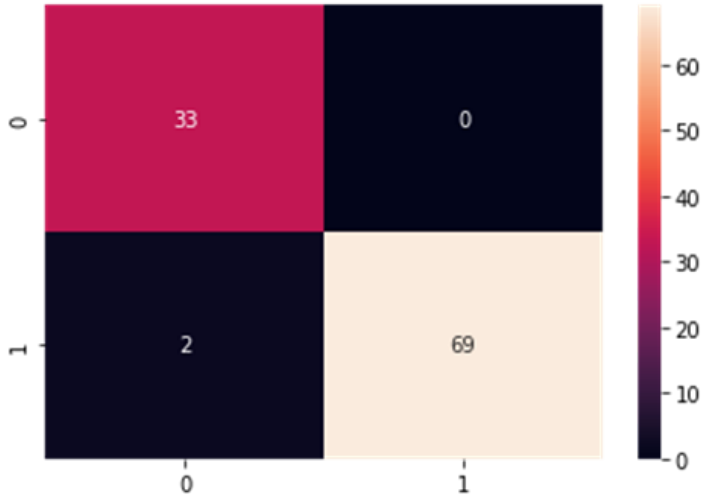
In **[Fig 12],** show the F1score is 98%, recall is 99% and precision 97%

	precision	recall	f1-score	support
0	0.97	0.94	0.95	33
1	0.97	0.99	0.98	71
accuracy			0.97	104
macro avg	0.97	0.96	0.97	104
weighted avg	0.97	0.97	0.97	104

[Fig 12]

5.1.12 Random Forest (RF)

Is the best model in algorithms with accuracy 98.08%. Get training score 100% and test score is 98.08% in **[Fig 13]** draw a plot of TP, TN and FP, FN in a confusion matrix.



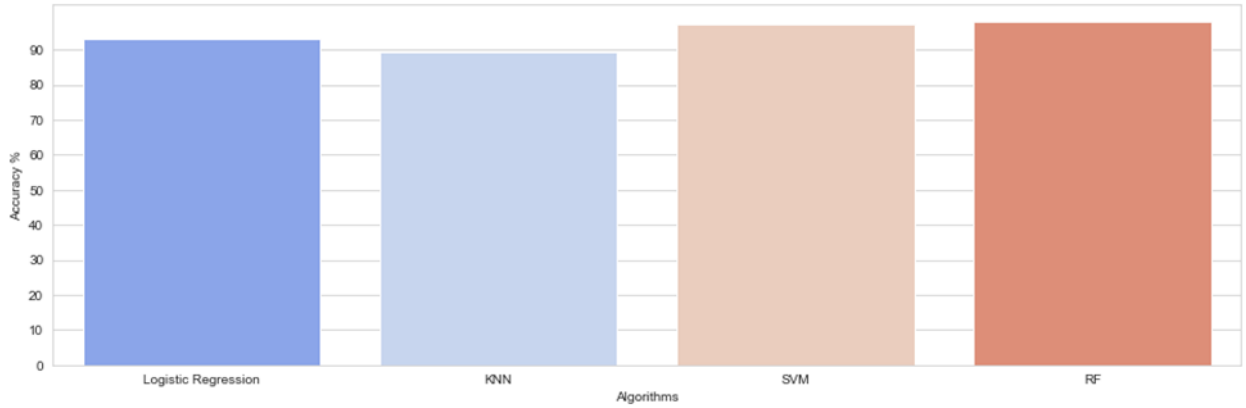
[Fig 13]

In **[Fig 14],** show the F1score is 99%, recall is 97% and precision 100%

	precision	recall	f1-score	support
0	0.94	1.00	0.97	33
1	1.00	0.97	0.99	71
accuracy			0.98	104
macro avg	0.97	0.99	0.98	104
weighted avg	0.98	0.98	0.98	104

[Fig 14]

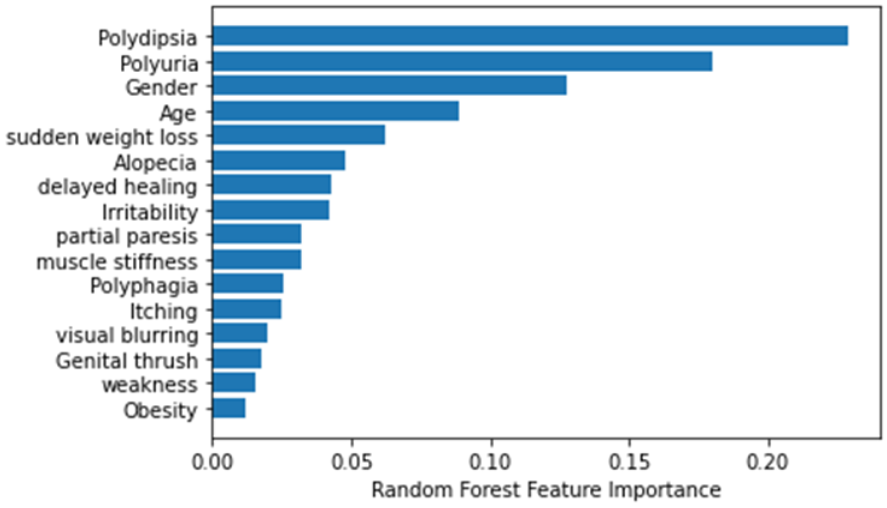
5.1.13 Accuracy algorithms



[Fig 15]

In above graph **[Fig 15]**, show the best algorithm in four algorithms. We found that random forest is the best algorithm with accuracy 98.08%, then choose the feature importance in highest algorithm accuracy and train the model on the feature importance to get the best result.

5.1.14 Feature Importance

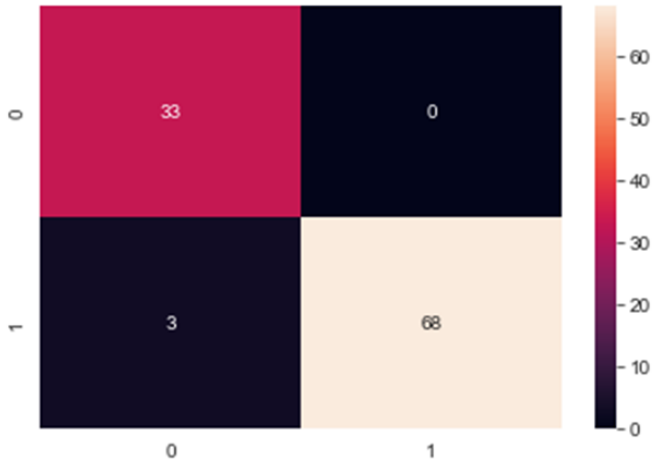


[Fig 16]

In the previous graph **[Fig 16]**, Show the important feature we found in the first sixth column is an important feature in the dataset. The model on this Sixth column includes: Polydipsia, Polyuria, Gender, Age, Sudden Weight loss, Alopecia.

5.1.15 Modify the model

In this modify the model with choose best feature and train model on these six features with random forest (RF). Get accuracy 97.12% and training score is 97.60% and test score is 97.11% in **[Fig 17]**, Show TP, TN and FP, FN in confusion matrix.



[Fig 17]

In Fig 18, show the F1score is 98%, recall is 96% and precision 100%

	precision	recall	f1-score	support
0	0.92	1.00	0.96	33
1	1.00	0.96	0.98	71
accuracy			0.97	104
macro avg	0.96	0.98	0.97	104
weighted avg	0.97	0.97	0.97	104

[Fig 18]

5.2 Classify Brain Tumors

5.2.1 What is a brain tumor?

A brain tumor is a growth of abnormal cells in the brain. The anatomy of the brain is very complex, with different parts responsible for different nervous system functions. Brain tumors can develop in any part of the brain or skull, including its protective lining, the underside of the brain (skull base), the brainstem, the sinuses and the nasal cavity, and many other areas. There are more than 120 different types of tumors that can develop in the brain, depending on what tissue they arise from.

In the United States, brain and nervous system tumors affect about 30 adults out of 100,000. Brain tumors are dangerous because they can put pressure on healthy parts of the brain or spread into those areas. Some brain tumors can also be cancerous or become cancerous. They can cause problems if they block the flow of fluid around the brain, which can lead to an increase in pressure inside the skull. Some types of tumors can spread through the spinal fluid to distant areas of the brain or the spine.

Context

Brain Tumors are complex. There are a lot of abnormalities in the sizes and location of the brain tumor(s). This makes it really difficult to completely understand the nature of the tumor. Also, a professional Neurosurgeon is required for MRI analysis. Oftentimes in developing countries the lack of skillful doctors and lack of knowledge about tumors makes it really challenging and time-consuming to generate reports from MRI’. So, an automated system on Cloud can solve this problem.

5.2.2 Dataset Information

Name of the dataset “Brain Tumor MRI” has collected images that are examined by the radiologist. A manual examination can be error-prone due to the level of complexities involved in brain tumors and their properties. Dataset is an image and contains two files: a training file and testing file and in these files contain four files (glioma tumor, no tumor, meningioma tumor and pituitary tumor). Total images is 3,264 split into two files training 2,870 images and testing 394 images.

Tools and IDE used:

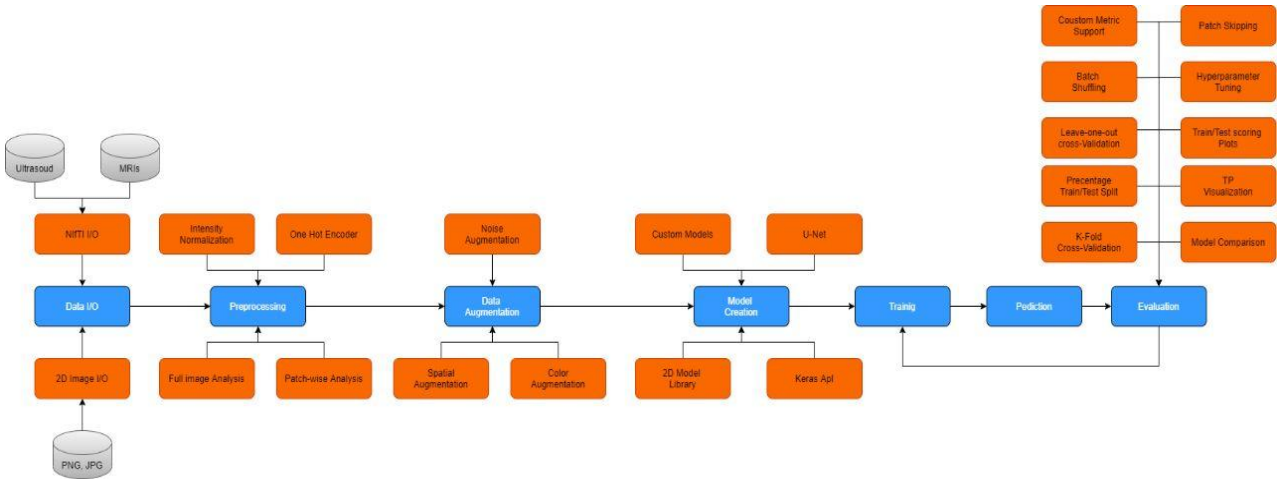
IDE: Kaggle with accelerator GPU T4 X2 (.ipynb file extension)

Programming Language: python

Package used:

- **Numpy**
- **Pandas**
- **OS**
- **Tqdm**
- **Cv2**
- **Matplotlib**
- **Seaborn**
- **PIL**
- **Tensorflow**
- **Keras**
- **Sklearn**

5.2.3 Deep learning Model flow



5.2.4 Data Preparation

1- Reshaping pictures to 150 X 150

2- Performing One Hot Encoding on the labels after converting it into numerical values

3- Resizing for the Deep learning model (making the last picture channel = 1)

5.2.5 Deep Learning Model

Using Layers (EfficientNetB0, Batch normalization, GlobalAveragePooling2D, Dropout, Flatten, Dense)

Dense is the output layer, which classifies the image into 1 of the 4 possible classes. It uses the softmax function, which is a generalization of the sigmoid function.

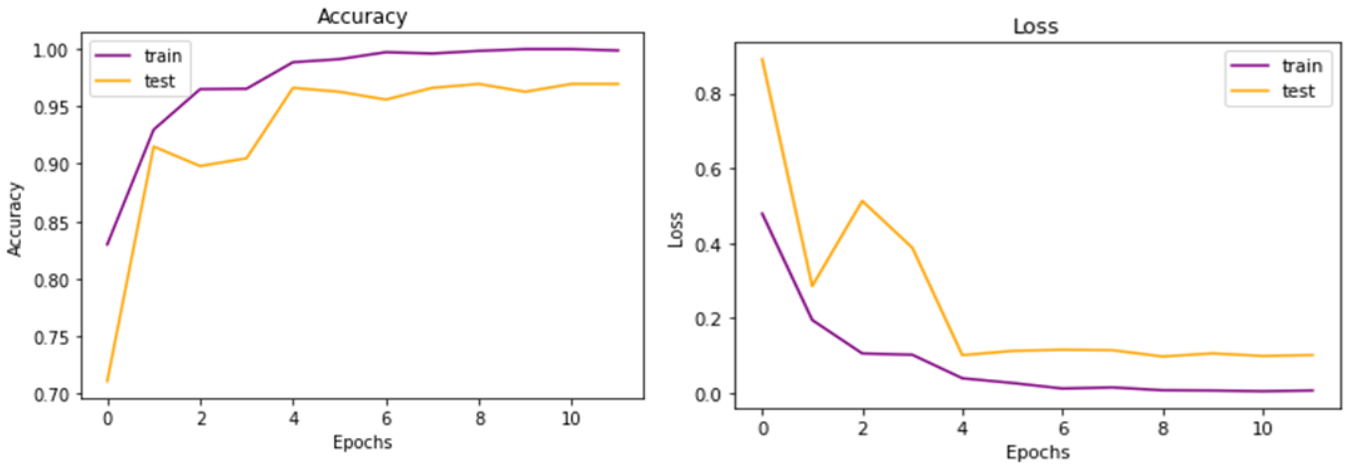
5.2.6 Transfer learning

A way to short-cut this process is to re-use the model weights from pre-trained models that were developed for standard computer vision benchmark datasets, such as the ImageNet image recognition tasks. Top performing models can be downloaded and used directly, or integrated into a new model for your own computer vision problems.

In this notebook, I will be using the EfficientNetB0 model, which will use the weights from the ImageNet dataset.

The include_top parameter is set to *False* so that the network doesn't include the top layer/ output layer from the pre-built model which allows us to add our own output layer depending upon our use case!

5.2.7 Model History

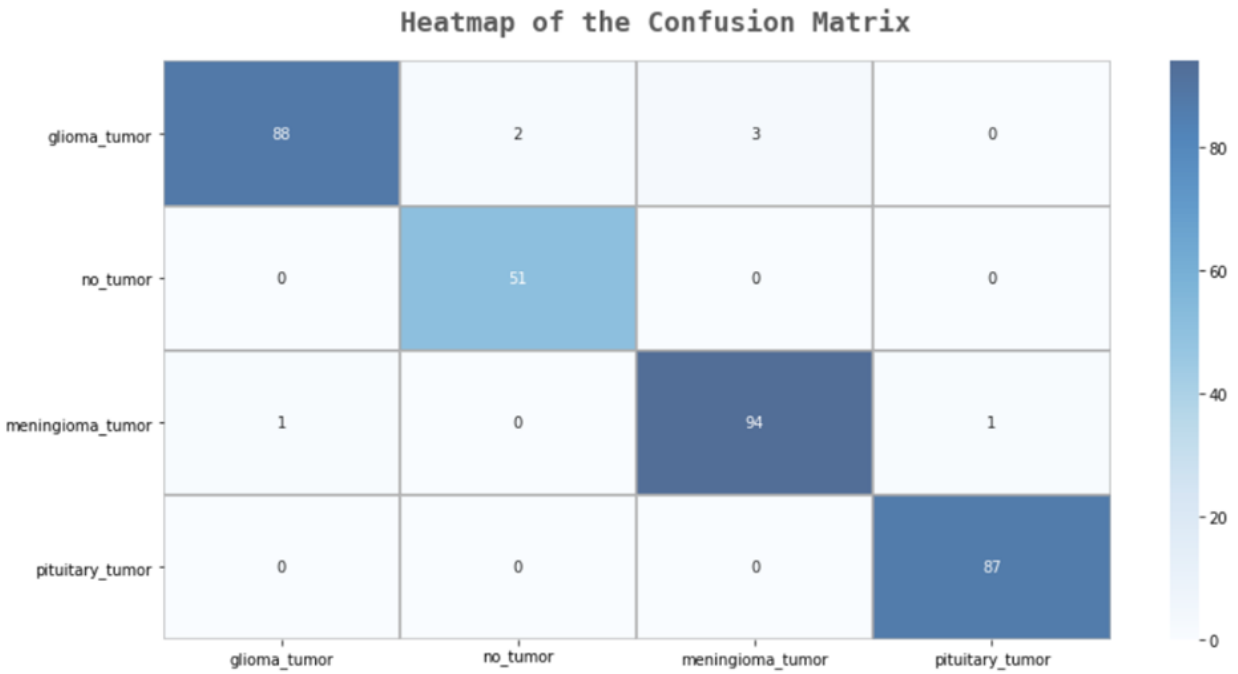


5.2.8 Evaluation Report

Test Accuracy: 98.165%

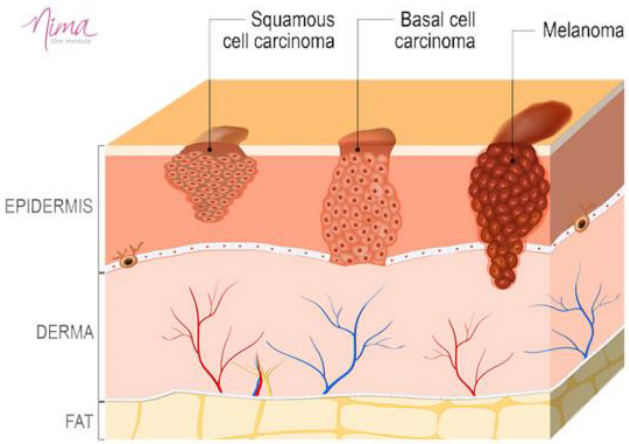
	precision	recall	f1-score	support
0	0.98	0.96	0.97	93
1	0.98	1.00	0.99	51
2	0.97	0.98	0.97	96
3	1.00	1.00	1.00	87
accuracy			0.98	327
macro avg	0.98	0.98	0.98	327
weighted avg	0.98	0.98	0.98	327

5.2.9 Confusion Matrix



5.3 Classify Skin Cancer

5.3.1 What is skin cancer?



Skin cancer is a type of cancer that develops in the skin cells. It is caused by the abnormal growth of skin cells, which can occur due to various factors, such as exposure to ultraviolet (UV) radiation from the sun or tanning beds, genetics, and environmental factors like exposure to certain chemicals.

There are three main types of skin cancer: basal cell carcinoma, squamous cell carcinoma, and melanoma. Basal cell carcinoma and squamous cell carcinoma are the most common types of skin cancer and are usually less dangerous than melanoma. However, if left untreated, they can still cause serious health problems.

Melanoma is a less common but more aggressive type of skin cancer that can spread to other parts of the body. It is often characterized by the appearance of a new mole or a change in an existing mole.

The best way to prevent skin cancer is to protect your skin from the sun by wearing protective clothing, using a broad-spectrum sunscreen with an SPF of at least 30, seeking shade during peak sun hours, and avoiding tanning beds. It is also important to perform regular skin self-examinations and to see a dermatologist if you notice any changes in your skin. Early detection and treatment of skin cancer can greatly improve the chances of a successful outcome.

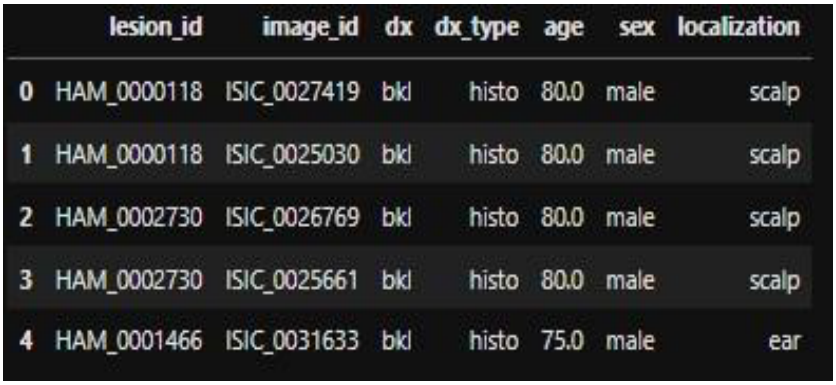
5.3.2 Context

About The Data:

Skin cancer affects many people in the world, where it's the most common of all cancer types. If a patient with a pigmented lesion could be identified as someone with or at a risk of developing skin cancer, then particular measures could be taken right away to lower their risk or destroy the cancer (if developed) at an early stage. These measures include immediate surgery, covering up the skin, limiting sun exposure, using a special type of sunscreen, avoiding tanning equipment, and drinking an extra amount of fluids.

5.3.3 Labeling Data :

Dataset include 7 different attributes:-



	lesion_id	image_id	dx	dx_type	age	sex	localization
0	HAM_0000118	ISIC_0027419	bkl	histo	80.0	male	scalp
1	HAM_0000118	ISIC_0025030	bkl	histo	80.0	male	scalp
2	HAM_0002730	ISIC_0026769	bkl	histo	80.0	male	scalp
3	HAM_0002730	ISIC_0025661	bkl	histo	80.0	male	scalp
4	HAM_0001466	ISIC_0031633	bkl	histo	75.0	male	ear

"lesion_id": ID of the patient
"imge_id": Unique ID for every image
"dx": Diagnostic skin cancer
"dx_type": Technical Validation field type
"age": Age of the patient
"sex": Gender of the patient
"localization": Location of skin Cancer in body

7 different diagnosis in data set:-

Melanocytic nevi (nv): benign neoplasms or hamartomas composed of melanocytes, the pigment-producing cells that constitutively colonize the epidermis (6705 Images).

Melanoma (mel): is a type of skin cancer that develops when melanocytes (the cells that give the skin its tan or brown color) start to grow out of control (1113 Images).

Benign keratosis-like lesions (bkl): Seborrheic keratoses are usually brown, black or light tan. The growths (lesions) look waxy or scaly and slightly raised (1099).

Basal cell carcinoma (bcc): Cancer that begins in the lower part of the epidermis (the outer layer of the skin) (514).

Actinic keratoses (akiec): A thick, scaly patch of skin that may become cancer (327).

Vascular lesions (vasc): relatively common abnormalities of the skin and underlying tissues, more commonly known as birthmarks (142).

Dermatofibroma (df): is a common cutaneous nodule of unknown etiology that occurs more often in women (115)

5.3.4 Data Size:

Before augmentation: 1015 as total images which have 80% to 20% split which are 8012 as train and 2003 as test.

After augmentation: 45756 as total images which have 80% to 20% split which are 36604 as train and 9152 as test.

5.3.5 Tools and IDE used:

IDE: Anaconda is most specifically Jupyter notebook (.ipyn file extension)

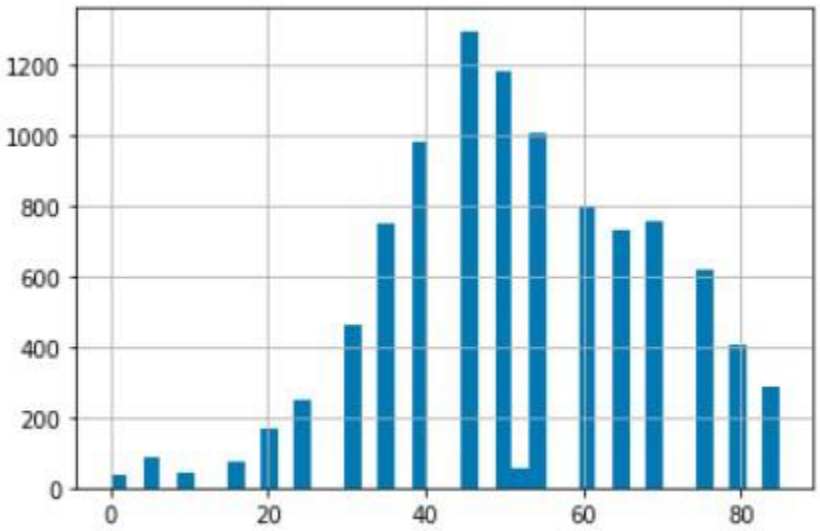
Programming Language: python

Packages used:

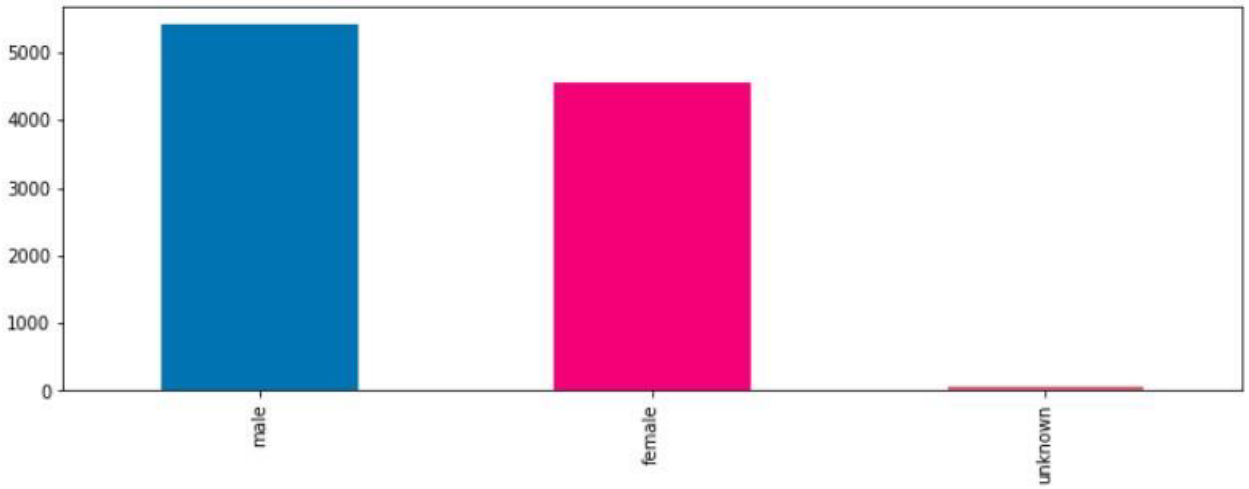
- **Numpy**
- **Pandas**
- **Matplotlib**
- **Seaborn**
- **Keras**
- **Sklearn**
- **cv2**
- **os**

5.3.6 Analysis:

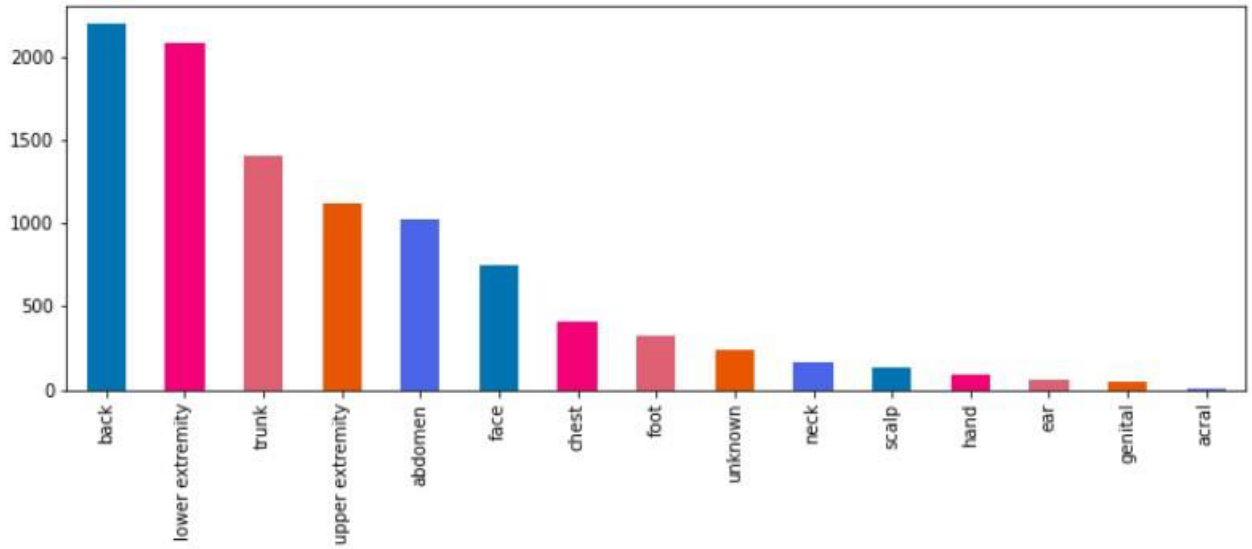
Patients age column



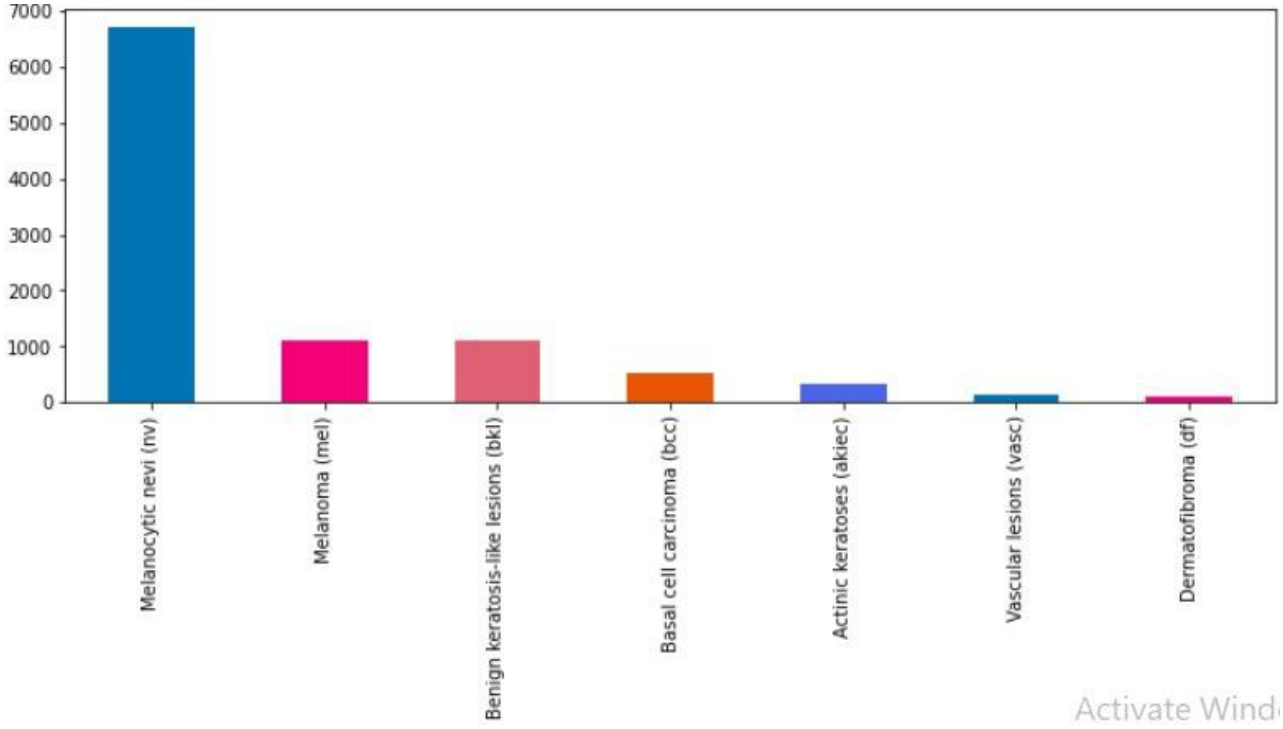
Gender



Cancer location on body



Cancer types



We had unbalanced classes and we solved it by augmenting data

5.3.7 Data Preparation:

1- Reshaping pictures to 28x28
2- Normalizing pictures (dividing pictures pixels by 255)
3- Data Augmentation (altering the data with small transformation to avoid overfitting)

5.3.8 Deep Learning Model:

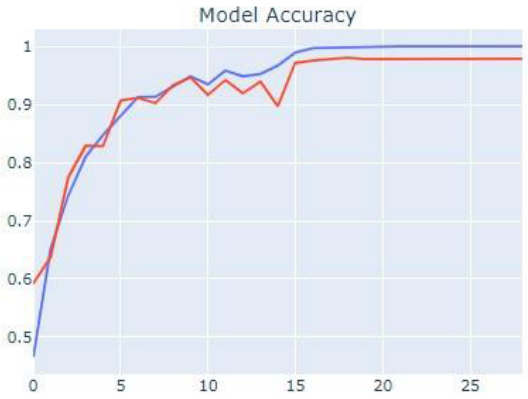
Using Layers (CNN 2D , Batch normalization , Max pooling2D , Dropout , Flatten , Dense)

With the last layer being a Dense layer with 7 nodes as it's activation function "Softmax" to output multi-class classification results

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 16)        448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 7, 7, 64)          18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 64)                32832
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 231
=================================================================
Total params: 132,583
Trainable params: 132,583
Non-trainable params: 0
```

5.3.9 Model History:

Training on 50 epochs with early stopping it stopped at the 29th epoch



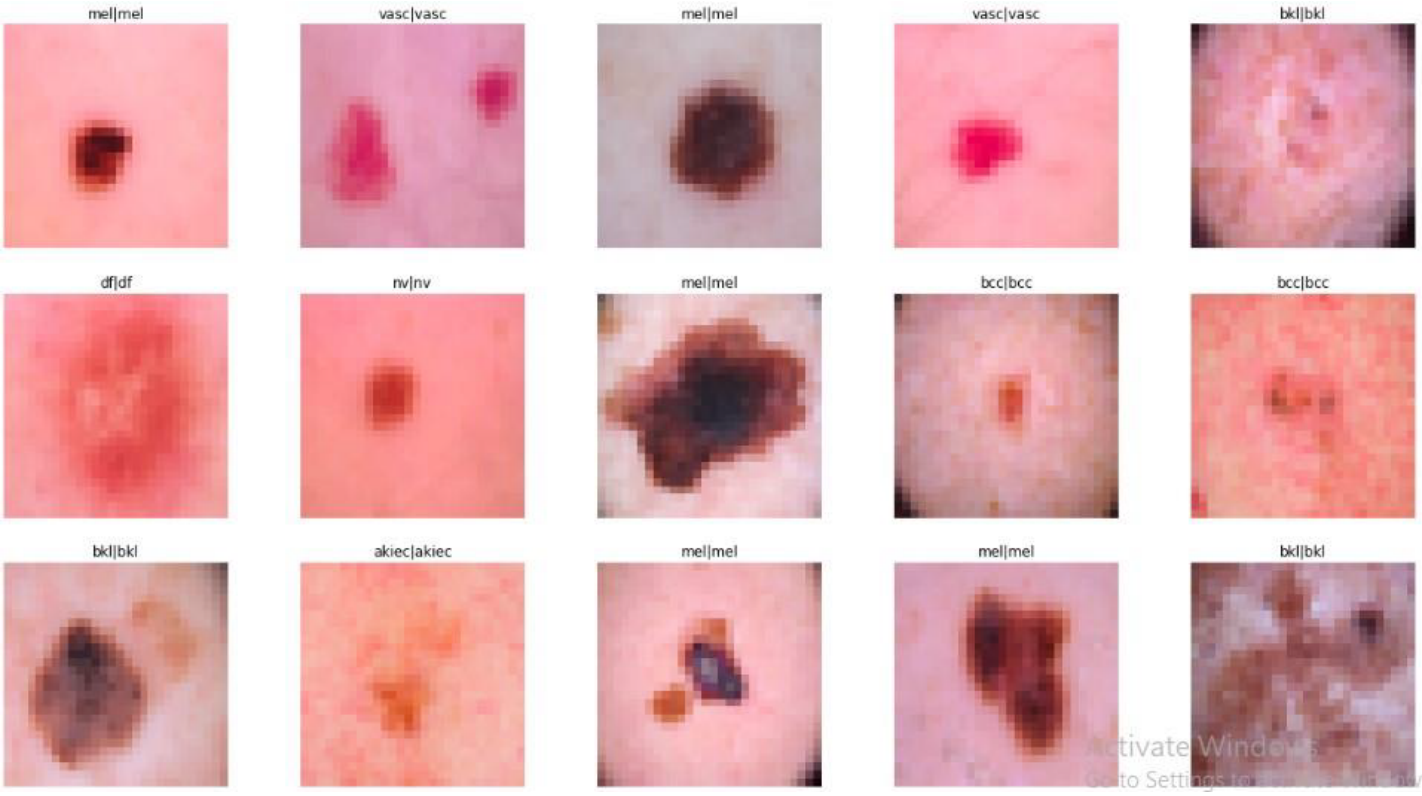
5.3.10 Test and Validation:

Testing the Model and predicting the Test Data which are 9152 pictures Model test

Test Accuracy: 98.022%

	precision	recall	f1-score	support
nv	0.99	0.88	0.93	1385
mel	0.94	0.99	0.97	1328
bkl	0.95	1.00	0.97	1294
bcc	0.99	1.00	1.00	1325
akiec	1.00	1.00	1.00	1270
vasc	1.00	1.00	1.00	1293
df	1.00	1.00	1.00	1257
accuracy			0.98	9152
macro avg	0.98	0.98	0.98	9152
weighted avg	0.98	0.98	0.98	9152

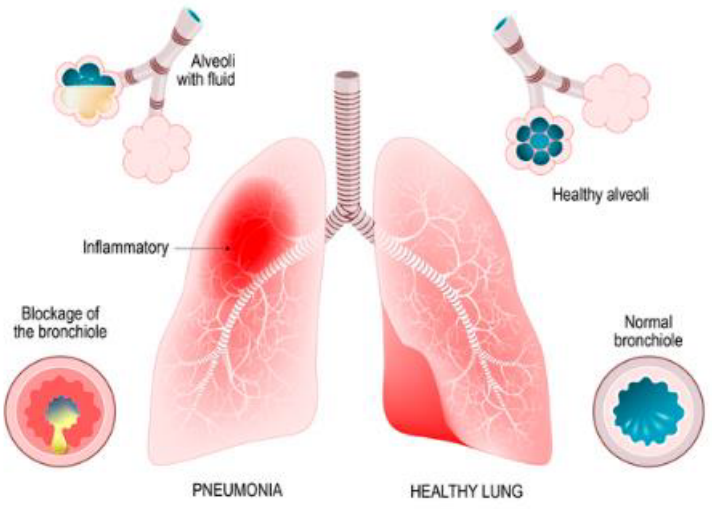
5.3.11 Model test in action:



5.4 Classify Pneumonia

Radiography

5.4.1 What is Pneumonia?

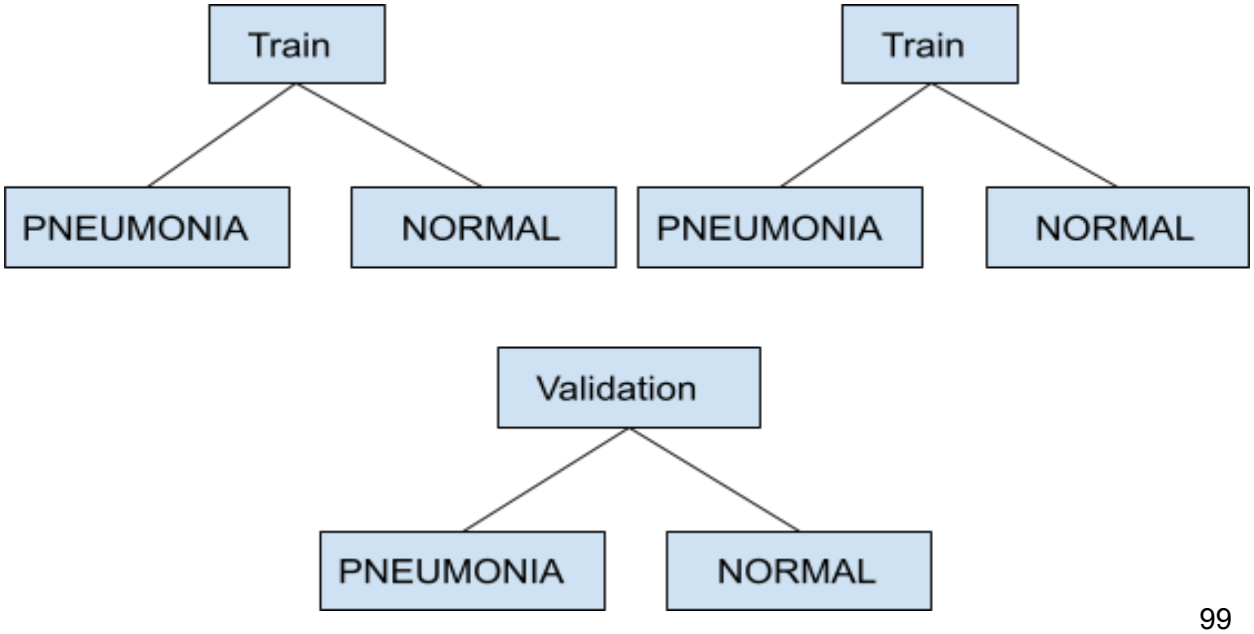


Pneumonia is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli.Symptoms typically include some combination of productive or dry cough, chest pain, fever and difficulty breathing. The severity of the condition is variable. Pneumonia is usually caused by infection with viruses or bacteria and less commonly by other microorganisms, certain medications or conditions such as autoimmune diseases.Risk factors include cystic fibrosis, chronic obstructive pulmonary disease (COPD), asthma, diabetes, heart failure, a history of smoking, a poor ability to cough such as following a stroke and a weak immune system. Diagnosis is often based on symptoms and physical examination. Chest X-ray, blood tests, and culture of the sputum may help confirm the diagnosis.The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated pneumonia.

5.4.2 Context :

According to the World Health Organization (WHO), pneumonia kills about 2 million children under 5 years old every year and is consistently estimated as the single leading cause of childhood mortality, killing more children than HIV/AIDS, malaria, and measles combined. The WHO reports that nearly all cases (95%) of new-onset childhood clinical pneumonia occur in developing countries, particularly in Southeast Asia and Africa. Bacterial and viral pathogens are the two leading causes of pneumonia but require very different forms of management. Bacterial pneumonia requires urgent referral for immediate antibiotic treatment, while viral pneumonia is treated with supportive care. Therefore, accurate and timely diagnosis is imperative. One key element of diagnosis is radiographic data, since chest X-rays are routinely obtained as standard of care and can help differentiate between different types of pneumonia. However, rapid radiologic interpretation of images is not always available, particularly in the low-resource settings where childhood pneumonia has the highest incidence and highest rates of mortality. To this end, we also investigated the effectiveness of our transfer learning framework in classifying pediatric chest X-rays to detect pneumonia and furthermore to distinguish viral and bacterial pneumonia to facilitate rapid referrals for children needing urgent intervention.

5.4.3 Labeling Data :



Three main files which are Train, Test, Validation each one of them have Two sub files containing label files which are PNEUMONIA which is considered as positive diagnosis and NORMAL which is considered as negative diagnosis with each sub file containing chest x-ray pictures of Each label.

5.4.4 Total images:

5216 - Train pictures
624 - Test pictures
15 - Validation pictures

5.4.5 Tools and IDE used:

IDE: Anaconda most specifically jupyter notebook (.ipyn file extension)

Programming Language: python

5.4.6 Packages used:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Keras
- Sklearn
- cv2
- Os

5.4.7 Data Preparation:

1- Reshaping pictures to 150 X 150
2- Normalizing pictures (dividing pictures pixels by 255)
3- Resizing for the Deep learning model (making the last picture channel = 1)
4- Data Augmentation (altering the data with small transformation to avoid overfitting)

5.4.8 Deep Learning Model:

Using Layers (CNN 2D, Batch normalization, Max pooling2D, Dropout, Flatten, Dense)

With the last layer being a Dense layer with one node as its activation function “Sigmoid” to output binary classification result (positive or negative)

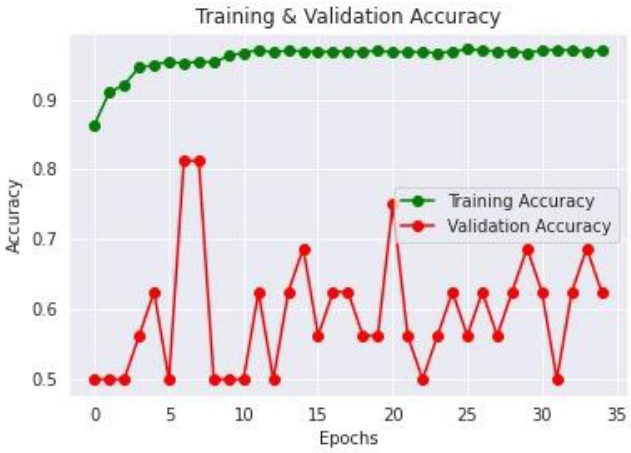
Model: "sequential_8"

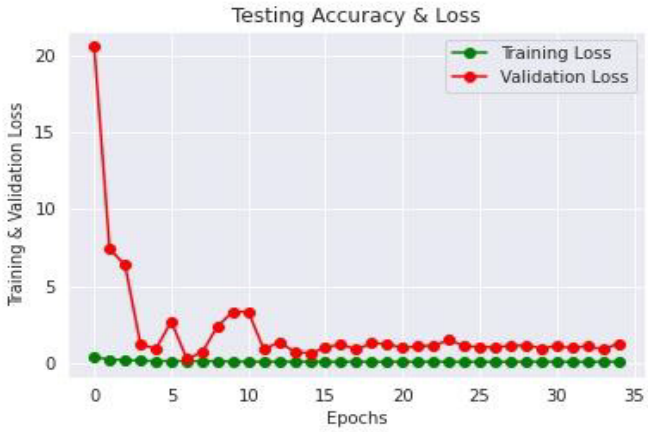
Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 150, 150, 32)	320
batch_normalization_36 (Batc	(None, 150, 150, 32)	128
max_pooling2d_36 (MaxPooling	(None, 75, 75, 32)	0
conv2d_37 (Conv2D)	(None, 75, 75, 64)	18496
dropout_29 (Dropout)	(None, 75, 75, 64)	0
batch_normalization_37 (Batc	(None, 75, 75, 64)	256
max_pooling2d_37 (MaxPooling	(None, 38, 38, 64)	0
conv2d_38 (Conv2D)	(None, 38, 38, 64)	36928
batch_normalization_38 (Batc	(None, 38, 38, 64)	256
max_pooling2d_38 (MaxPooling	(None, 19, 19, 64)	0
conv2d_39 (Conv2D)	(None, 19, 19, 128)	73856
dropout_30 (Dropout)	(None, 19, 19, 128)	0
batch_normalization_39 (Batc	(None, 19, 19, 128)	512
max_pooling2d_39 (MaxPooling	(None, 10, 10, 128)	0
conv2d_40 (Conv2D)	(None, 10, 10, 256)	295168
dropout_31 (Dropout)	(None, 10, 10, 256)	0
batch_normalization_40 (Batc	(None, 10, 10, 256)	1024
max_pooling2d_40 (MaxPooling	(None, 5, 5, 256)	0
flatten_8 (Flatten)	(None, 6400)	0
dense_15 (Dense)	(None, 128)	819328
dropout_32 (Dropout)	(None, 128)	0
dense_16 (Dense)	(None, 1)	129

Total params: 1,246,401
Trainable params: 1,245,313
Non-trainable params: 1,088

5.4.9 Model History:

Training on 35 epochs





5.4.10 Test and Validation:

Testing the Model and predicting the Test Data which are 624 pictures

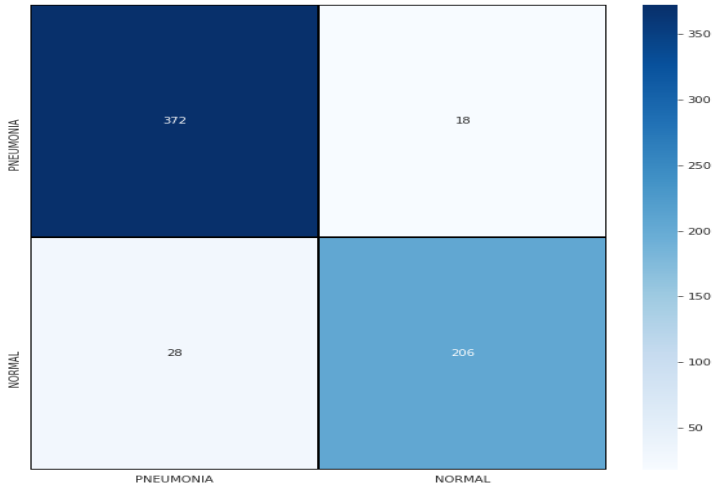
```
624/624 [==============================] - 0s 734us/step
Loss of the model is -  0.25089872704866606
624/624 [==============================] - 0s 672us/step
Accuracy of the model is -  92.62820482254028 %
```

Using Classification scoring functions

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.93	0.95	0.94	390
Normal (Class 1)	0.92	0.88	0.90	234
accuracy			0.93	624
macro avg	0.92	0.92	0.92	624
weighted avg	0.93	0.93	0.93	624

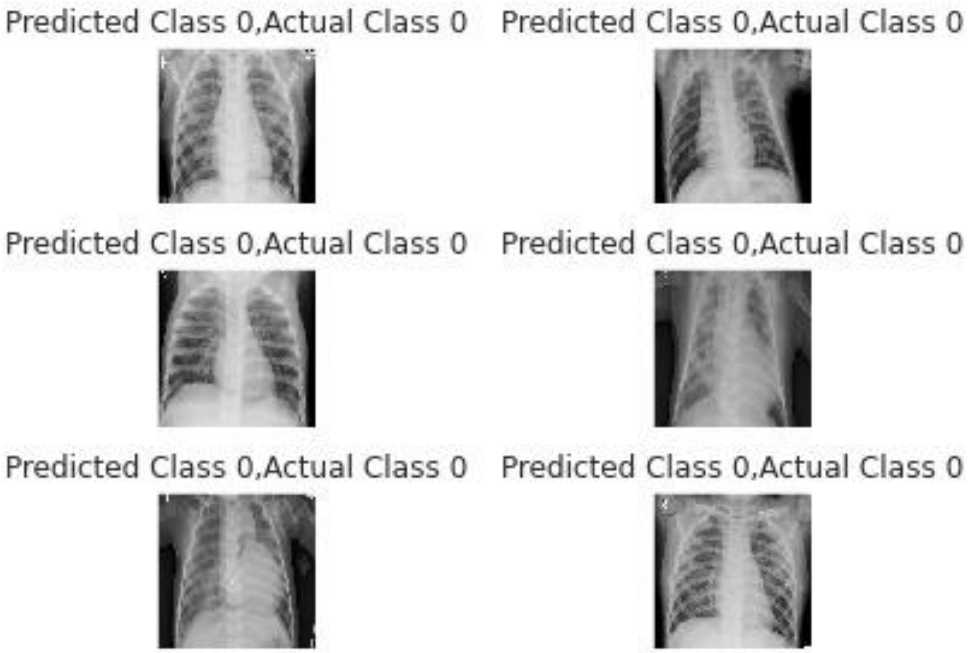
Precision: percentage of True positives the model got right from the actual results
Recall: percentage of True positives the model got right from the predicted results

5.4.11 Confusion Matrix



5.4.12 Model Rights and Wrongs examples:

Model predicted Right:



Model predicted wrong: